INTEGRATING COMPUTATIONAL LEARNING IN PROBABILITY

Amy S. Wagaman
Department of Mathematics and Statistics
Amherst College, Amherst, MA 01007
awagaman@amherst.edu

The mathematical foundations of probability can be challenging for our students to learn, and our students tackle many problems for practice. While analytical solutions to some problems can be difficult, empirical simulations can give intuition and guidance to students, provided that the students are able to perform the simulations. We discuss integrating computational learning in a probability course with a goal to strengthen student knowledge of probability concepts, algorithmic thinking, and computational skill. These skills also assist with bridging the gap between statistical theory and practice. Specific computational skill goals include writing functions, writing simulations to verify analytical results (including communicating results), and using a reproducible workflow.

BACKGROUND

A course in Probability theory is typically expected of undergraduate statistics majors and is of interest for students interested in mathematics, economics, physics, chemistry, and many other disciplines. These courses may cover both discrete and continuous distributions, or there may be separate courses so that students can tackle discrete distributions with less of a calculus pre-requisite. Probability courses tend to cover topics such as counting methods, expectation, variance, common probability distributions, working with functions of a random variable, leading up to limit theorems and other results. Our focus is not on this content, but rather, we want to consider the computational learning that goes along (or should go along) with this content.

The ASA Curriculum Guidelines (2014) detail the need for statistics students to be able to program, perform algorithmic problem solving, and use simulation-based techniques. Computational skills required include data wrangling, working with databases, and writing simulations, after which, results must be well-communicated. Thus, the Guidelines suggest we should consider the computational and communication aspects in our courses. The more recent Guidelines for Programs in Data Science (De Veaux et al., 2017) include computational and statistical thinking, and algorithms and software foundation as two of the six key competencies for an undergraduate data science major. The computational and statistical thinking competency reflects the argument from Breiman (2001) about the two cultures of algorithmic and data models. The increased focus on computational expectations in both sets of guidelines is clear, but we need to work to integrate these concepts into our courses.

Exploration of computational tools in statistics courses has a long history. Mills (2002) examines over 40 references that use computer simulation methods to teach statistics (in a variety of fields and for a wide range of topics). Some focus has been on the Probability course. For example, Dinov & Sanchez (2006) explore the use of interactive applets in a Probability course to improve understanding of probability concepts. Pfannkuch et al. (2016) take a different approach, garnering insights into teaching probability (at several levels) by interviewing individuals who use probability models on a daily basis in their work. The focus of these works is on different teaching methods, content framework, and/or new technology, not necessarily trying to lay out the computational framework or skills that students should be developing while in a Probability course, or study how effectively students can learn those computational skills. We want to examine that set of skills more closely.

At our institution, our Probability course is a pre-requisite for several other courses, including our Theoretical Statistics course (which is required for our statistics majors). In Theoretical Statistics, students engage in empirical simulations to verify analytical results and solve problems (Horton 2013). If that course is their first introduction to writing their own simulations, the students can be overwhelmed with trying to learn the computational and theoretical material. To help ameliorate this issue, we have been working to scaffold more computational learning into our Probability course (as well as our earlier courses), and require our

In M. A. Sorto, A. White, & L. Guyot (Eds.), *Looking back, looking forward. Proceedings of the Tenth International Conference on Teaching Statistics (ICOTS10, July, 2018), Kyoto, Japan.* Voorburg, The Netherlands: International Statistical Institute. iase-web.org

majors to take more computer science courses. Our current computational skill goals for Probability students include understanding and writing functions, understanding how to set up and write simulations to verify results, and using a reproducible workflow.

IMPORTANCE

At a time when the statistics curriculum is still undergoing rapid development, with the rise of data science and computational expectations, it is important to consider the role of computation in courses that have traditionally been more theoretical in nature. By adding computational learning goals to Probability, we believe we are better preparing students for additional coursework in many fields and improving their ability to tackle real-life computational problems, and providing connections to other courses such as those in Data Science. The era of probability tables is over – our students need to be facile with appropriate technology.

For the particular computational learning goals we have in mind, we believe that students who learn to write their own functions are better able to understand functions written by others and can use functions to streamline their workflow, as well as learn to break a problem down into smaller pieces (algorithmic thinking). Being able to write a simulation can help students check their understanding and intuition on a problem. Indeed, in some circumstances, simulations may be easier to do than direct derivations, and may help provide guidance and intuition for analytic solutions. Setting up an appropriate simulation (and writing necessary or useful functions) develops foundational algorithmic thinking skills. Finally, a reproducible analysis tool can help scaffold learning.

COURSE MATERIALS AND ACTIVITIES

In recent semesters of our Probability course, there has been a greater emphasis on computational learning and algorithmic thinking as part of the learning goals. Briefly, we describe what this looked like from our perspective as the course instructor, and will address student comments and results in a later section.

Our course regularly enrolls over 30 students (at a small liberal-arts institution) of varied backgrounds. Students are introduced to R (2009) the first week (though many have seen it before), and we use textbooks for the class that are compatible with R or use R directly. The textbook used in the most recent iterations of the course is Dobrow (2013), which includes R throughout. Our specific implementation of R in class is an RStudio (2011) server where the students work with RMarkdown (2014) files, so that they can interweave text and code.

Students engage with the software in weekly lab activities written to supplement the course material. These labs range from learning to graph probability distributions and finding probabilities with commands to exploring a simulation and using it to address a problem. Each homework is also accompanied by a problem that includes a computational component that requires the use of the software. The computational components of the homework assignments increase in difficulty over the course of the semester. For example, the first week, students learn the basic components of a function in R and learn to tweak one that is provided to create a new function to do a very simple task. They also learn about pseudocode (an outline of appropriate steps for a simulation) and are prompted to break the problem down into the steps they need to accomplish on practically every assignment. In the third week, they have to write their own function, and use it to explore a few settings (i.e., starting to simulate). A few weeks later, they are presented with a problem and tasked with writing a simulation to arrive at a solution. At least three of the assignments involve students writing their own simulations to verify results, and in one of those cases, the simulation was much easier for the students to grasp than the analytical solution.

Providing this scaffolding in the homework assignments allowed students to work on their computational and algorithmic thinking skills while still tackling all of the probability content. The structured development meant that students always had previous similar examples to build on, but the increasing difficulty in assignments posed an appropriate challenge.

Our exams included at least one problem requiring students to generate pseudocode or read output from a simulation (process what the code is doing and make use of the output). In addition, students have a course project that has a computational component that requires writing a simulation, and using a function to address a question of interest. Students then had to

communicate their results in a few paragraphs, using the simulation as support for their conclusions.

In the most recent iteration of the course (Fall 2017), students were also provided with RSupplement files, written by me, which took the R code from the textbook and supplemented it with detailed comments about what each step in the code was doing, as well as including some additional material not included in the text – additional example simulations or graphics to illustrate concepts, for example. These were provided as RMarkdown documents, and students were instructed to read them along with the textbook material for each class meeting. Students could also run the code in the RSupplements and experiment with changing values. This was designed to increase student interaction with the software and code, as we envisioned that not many students were stopping their reading to open R, typing in the example code from the textbook, and observing the results. It also provided the students with additional examples to reference when working on their assignments.

SAMPLE ACTIVITY

This sample activity serves to illustrate a typical mid to late semester course computational homework problem with scaffolding (completed in addition to other problems from the textbook). Students are provided with an RMarkdown template with the following components. My comments about each part are provided in italics (and were not provided to the students).

a. Solve Problem 5.6. (Dobrow, 2013, *205*) *This is a variant of the coupon collector problem, with 15 "professors" to collect by taking classes. Students should have been able to solve this following the book examples.*

b. Now suppose Tina only takes 10 courses in the math department. What is the expected number of different professors she will have? (As in part a, assume that every time Tina takes a course, each professor is equally likely to be the instructor). *There is a similar example to this in the text as well, but students have to recognize the context. For many students, a simulation is easier to grasp than the analytical solution here.*

We want to simulate and verify the results in part b. Recall that there are usually many ways to accomplish a programming task in R, so you might approach this differently than your classmates.

c. Provide pseudocode to outline a reproducible simulation to verify your results in part b. *This step is required so that students think about what they need to do before trying to do it. If students ran into issues with code that related to the algorithmic process for the simulation, I asked to see their pseudocode. If they hadn't written anything here, I instructed them they had to do that first.*

d. Provide the R code for your reproducible simulation and run it. *Students accomplished this in many ways. Some students wrote functions, others just wrote a loop or used the replicate() or mosaic::do() functions in R.*

e. Write a few sentences to compare the results of your simulation to your computations in part b. *This step is necessary so that students learn to pull necessary information from the simulation, and learn to communicate their findings, as well as learn to validate their own results.*

ASSESSMENT

In order to assess the impact of our use of the RSupplements and the scaffolding structure for the assignments (as well as the related writing components of the course), we opted to solicit student feedback via surveys. The study was approved by the Amherst College Institutional Review Board. Students who opted in to the study (all 31 students) were given a short survey mid-semester, and a second, longer survey at the end of the course.

For the mid-semester survey, the focus was on the RSupplements, as those were new to the course. Students were asked to rate their level of agreement (SD=strongly disagree, D=disagree, N=Neutral, A=agree, SA=strongly agree) with the following statement: "I have engaged with the

RSupplements." Students were also asked to "Indicate one aspect of the R Supplements that has benefitted your computational learning for Probability." and to "Indicate one way that you feel the RSupplements could be improved or used to better support your computational learning in Probability." 26 of 31 students submitted replies to this survey.

The end of semester survey included questions covering the following aspects of the course: writing, computation, the RSupplements, and the feedback received on submitted assignments about writing and computation. For our purposes, we will focus on the questions associated with the computational portions of the homework (R portion) and the RSupplements. Students were asked to rate their level of agreement (using the same scale as above) with the following statements in relation to the writing on the R portion of the Homework assignments:

- The assignments helped me understand my own thought processes in tackling the problems.
- The assignments helped me understand how to present the steps of my probabilistic thinking in the solution.
- The feedback on the assignments helped me improve my ability to communicate probabilistic/statistical concepts.

Students were asked to rate their level of agreement with the following statements in relation to the RSupplements:

- I read the RSupplements (as assigned, in conjunction with the textbook sections) each week.
- The RSupplements helped me engage with and learn some R.
- The commentary and extra material provided in the RSupplements was beneficial to my learning how to use R for probability.

The two open-ended questions from the mid-semester survey were repeated here as well. 21 of 31 students submitted replies to this survey. Of those, 4 students completed the end of semester survey but did not fill out the mid-semester survey. Thus, 17 of the 31 students completed both surveys.

RESULTS

First, we will convey our sense of the outcomes for the students with this increased computation in the course. Then, we present the student responses in relation to the RSupplements, computational outcomes, and related writing.

Overall, the increased computation in Probability seemed to serve the students well. They seemed to grasp the idea of a reproducible workflow and were able to work with the probability distributions via R commands fairly easily. Simulations were a bit more challenging, but with support, all students were able to complete the homework assignments and have working simulations. At a minimum, students advancing into our Theoretical Statistics course had exposure to writing their own (working) simulations, and communicating results. The projects had less scaffolding (individual assignment, no collaboration allowed), and simulations caused issues for one student there (it was a working, but incorrect, simulation). All students were able to address the final questions in the project that relied on a provided function where they had to explore the parameter space of the function inputs and use the results to support their findings. Overall, it seemed students were able to write functions and use them to tackle appropriate problems, although writing pseudocode for exams proved challenging for some students.

To frame the survey results, we consider how students responded to their level of engagement with the RSupplements on the mid-semester survey, and how they responded to whether they read them as assigned on the end of semester survey (though those questions are not exactly equivalent). For the mid-semester survey, 17 out of the 26 students were in agreement with the statement (Agree -14, Strongly Agree - 3). The remainder of the responses were: Neutral – 6, Disagree – 2, and Strongly Disagree -1. By the end of the semester, more students responded that they were in disagreement or neutral with the associated statement, which seems to imply that while they used the RSupplements early on, they either did not keep up with the reading or did not feel a need to refer to them.

In terms of responses to whether the RSupplements were useful for learning R, it is important to keep in mind the class make-up. Of the 31 students in the course, roughly half were statistics majors (or students with intention to declare in statistics). Thus, roughly half the class had previous exposure to R from at least one course (if not two or three courses). The responses to this

survey question indicate that of the 21 respondents, 10 were in agreement with the statement (Agree – 8, Strongly Agree – 2) with the remaining 11 either Neutral (7), or in disagreement with the statement (Disagree – 3, Strongly Disagree -1). This is not surprising given the class makeup. A better question would have been to ask whether the RSupplements conveyed the importance of the algorithmic thinking in their computational content, rather than inquiring about learning the software.

To assess if the RSupplements were useful to the students, we consider the responses from whether the students felt that the material was beneficial to learning how to use R for probability. Of the 21 responses, 12 students agreed with the statement (Agree - 9, Strongly Agree - 3), with 6 being Neutral, and 3 responding Disagree. Overall, the sentiment seemed to be that the material was useful. Specific comments from the open-ended questions about how they were beneficial included repeated comments that they were useful for the homework, for examples of simulations, and for detailed explanations of the steps in said simulations.

Clearly though, student interaction with the material was not as high as I would have liked given the set of computational learning outcomes, and students had suggestions about how to increase their interaction with the RSupplements. These included suggestions to refer to them more in class (I only walked through one or two examples using them during the semester), assign extra credit problems that required them, make the homework problems harder so they'd be more necessary to reference, as well as comments about their length needing to be trimmed (when combined with course workload).

To look at the impact of the assignment scaffolding on the outcomes of learning to write functions and simulations, we can examine the responses about the R portions of the homework. First, we examine responses to the statement: "The assignments helped me understand my own thought processes in tackling the problems." Of the 21 replies, 16 were in agreement with the statement (Agree – 10, Strongly Agree – 6) with the remaining 5 being Neutral (3) or Disagree (2). This suggests that the scaffolding and encouraging pseudocode was beneficial.

Next, we consider whether students thought the assignments were useful in learning how to present the steps of a probabilistic argument (their probabilistic thinking). Of the 21 replies, 15 were in agreement with the statement (Agree -11, Strongly Agree – 4), with 6 being Neutral (4) or Disagree (2). The Disagree responses shared one student between this and the previous statement. Notably, one student thought the assignments were useful for understanding their own thought processes but not learning to present the steps of that thinking.

Finally, we consider the student responses to the feedback on the assignments, in terms of learning to communicate probabilistic and statistical concepts. Here, there were 14 replies in agreement (Agree – 11, Strongly Agree – 3), with the remaining 7 being Neutral (5) and Disagree (2). While still overall positive, it appears that I should consider refinements to my feedback. Providing feedback on these assignments included comments on the writing (clearly communicating results), the code, and the probability/statistical aspects of their explanations. This is a challenging trio, with the writing aspect potentially particularly challenging for statistical educators not used to providing feedback about it.

Future work includes examining student mastery of the computational learning goals that were added to the course. Overall the use of supplemental material to assist with learning and scaffolding of the assignments appeared to be useful for many students.

CONCLUSION

We have described changes to our Probability course in an attempt to satisfy new computational learning expectations for our students. We argue that these expectations are in line with new guidelines for statistics and data science students (ASA Curriculum Guidelines, 2014; De Veaux et al., 2017). While we have not yet undertaken research to see how well students are mastering those learning expectations, we have presented results here about how our approach was received by students and their assessments of the provided course materials, as well as anecdotal evidence of their mastery. The supplementary materials with additional explanations about aspects of the code and purpose of the commands seemed beneficial to students, and some suggestions were made for future improvements.

The computational goals of interest were to understand and write functions, understand how to set up and write simulations to verify results, and undertake a reproducible workflow. Students seemed to find the scaffolding approach of assignments (particularly the homework problem with a computational component) useful to understand their own thinking about the problem and structure a response including a probabilistic argument. As evidenced over the course of the semester, writing simulations was the hardest of these learning goals for students to achieve. Future work to examine how this skill is taught in other fields, such as computer science or physics, might be enlightening.

Providing our statistics students with skills that are both needed and which supplement their learning should be a high priority for statistics educators. Computational software can do amazing things, but only if one knows how to break down a problem and request the software work through it well. Adding computational learning goals to a course can feel challenging – it is more to teach after all, but the benefits for our students are immense. To conclude, one student responded to an open-ended end of the semester survey question with the following summary statement about what they learned during the semester in relation to the computational aspects of the course: "pseudocode -> easier life."

REFERENCES

Breiman, L. (2001). Statistical Modeling: The Two Cultures. *Statistical Science, 16* (3), 199-231.

ASA Guidelines Group. (2014). ASA Curriculum Guidelines for Undergraduate Statistics Programs, www.amstat.org/asa/education/Curriculum-Guidelinesfor-Undergraduate-Programs-in-StatisticalScience.aspx.

De Veaux, R.D., et al. (2017). Curriculum guidelines for undergraduate programs in data science, *Annual Review of Statistics and its Applications*. DOI:10.1146/annurev-statistics-060116-053930.

Dinov, I., & Sanchez, J. (2006). Assessment of the pedagogical utilization of the statistics online computational resource in introductory probability courses: A quasi-experiment. International Conference on Teaching Statistics 7.

Dobrow, R. P. (2013). *Probability: with applications and R*. Hoboken, New Jersey: John Wiley & Sons.

Horton, N. J. (2013). I hear, I forget. I do, I understand: a modified Moore-method mathematical statistics course. *The American Statistician, 67*(4), 219-228.

Mills, J. D. (2002). Using computer simulation methods to teach statistics: A review of the literature. *Journal of Statistics Education, 10*(1).

Pfannkuch, M., Budgett, S., Fewster, R., Fitch, M., Pattenwise, S., Wild, C., & Ziedins, I. (2016).
Probability modeling and thinking: What can we learn from practice? *Statistics Education Research Journal, 15*(2).

R Development Core Team. (2009). R: A language and environment for statistical computing. *R Foundation for Statistical Computing*. Vienna, Austria. ISBN 3-900051-07-0.

RStudio. (2011). RStudio, new open-source IDE for R. *RStudio Blog*. http://blog.rstudio.org/2011/02/28/rstudio-new-open-source-ide-for-r/.

RStudio. (2014). R Markdown v2. *RStudio Blog*. http://blog.rstudio.org/2014/06/18/r-markdown-v2/, last accessed February 14, 2018.