Integrating Computational Learning in Probability

Amy S. Wagaman
Amherst College
Amherst, MA, USA

Outline

- Background
- Computational Learning Goals
- Computation in the Course
- Example Assignment
- Assessment

A Typical Probability Course

- Expected of undergraduate statistics majors
- Of interest for students interested in mathematics, economics, computer science, and other disciplines
- Cover topics such as:
  - counting methods,
  - expectation and variance,
  - common probability distributions,
  - working with functions of a random variable,
  - leading up to limit theorems and other results.
- Our focus is on the **computational learning** that goes along with this content.

Where is the Computation?

- ASA Curriculum Guidelines (2014) says statistics students need to be able to:
  - program,
  - perform algorithmic problem solving,
  - use simulation-based techniques.

- Guidelines for Programs in Data Science (2017) include:
  - computational and statistical thinking (reflects Breiman (2001)),
  - algorithms and software foundation

Why Teach Computation?

- Rise of data science and computational expectations
- The era of probability tables is over – our students need to be facile with appropriate technology.
- By adding computational learning goals to Probability, we believe we are:
  - better preparing students for additional coursework in many fields,
  - improving their ability to tackle real-life computational problems,
  - providing connections to other courses such as those in Data Science.

Computational Learning Goals

- Focus on algorithmic thinking and computational competency
- Specifically:
  - understanding and writing functions,
  - understanding how to set up and write simulations to verify results,
  - using a reproducible workflow.

- These are in addition to the usual course learning goals for probability content.

About Amherst

- Relatively small liberal arts institution (1800 undergraduates)
- Amherst Statistics major is fairly new with our fourth set of graduates in 2018.
- Probability course is taught by a statistician (since 2005, roughly).
- Course counts as an elective for mathematics majors and is required for the statistics major.
- Kyoto Connection: Doshisha University is a sister institution

Course Background

- Met twice weekly for 110 minutes each
- Enrolls slightly over 30 students of varied backgrounds.
- Students are introduced to R (2009) the first week (about half have seen it before)
- Textbooks are compatible with R or use R directly.
  - Most recently, Dobrow (2013), which includes R throughout.
- Use an RStudio (2011) server where the students work with RMarkdown (2014) files, so that they can interweave text and code.

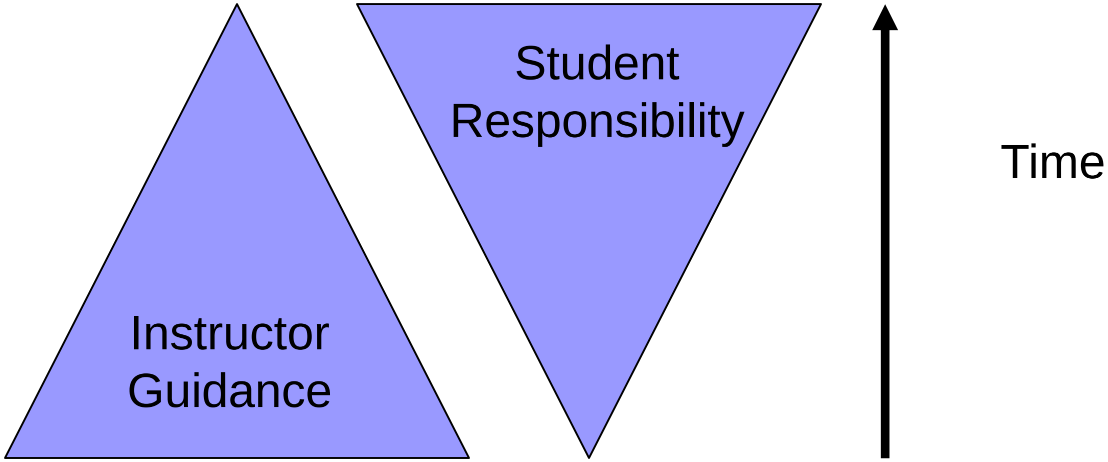
Course Materials and Activities

- How students engaged with computation:

Assignment/Activity	How Often
Lab activities in class	At least once a week
Homework – additional problem	Nine of ten assignments
Writing a solution to probability questions	Three times in the semester
Exams (two midterms and final)	One computational problem on each
Project (used R and required simulation)	Final course project
R Supplements (reading) + Textbook	Assigned for every class

Scaffolding

- Using scaffolding allows students to work on their computational and algorithmic thinking skills while still tackling all of the probability content.
- Build up basic skills with computation with guidance, gradually remove guidance (still provide support) so students are solving problems on their own.



Scaffolding

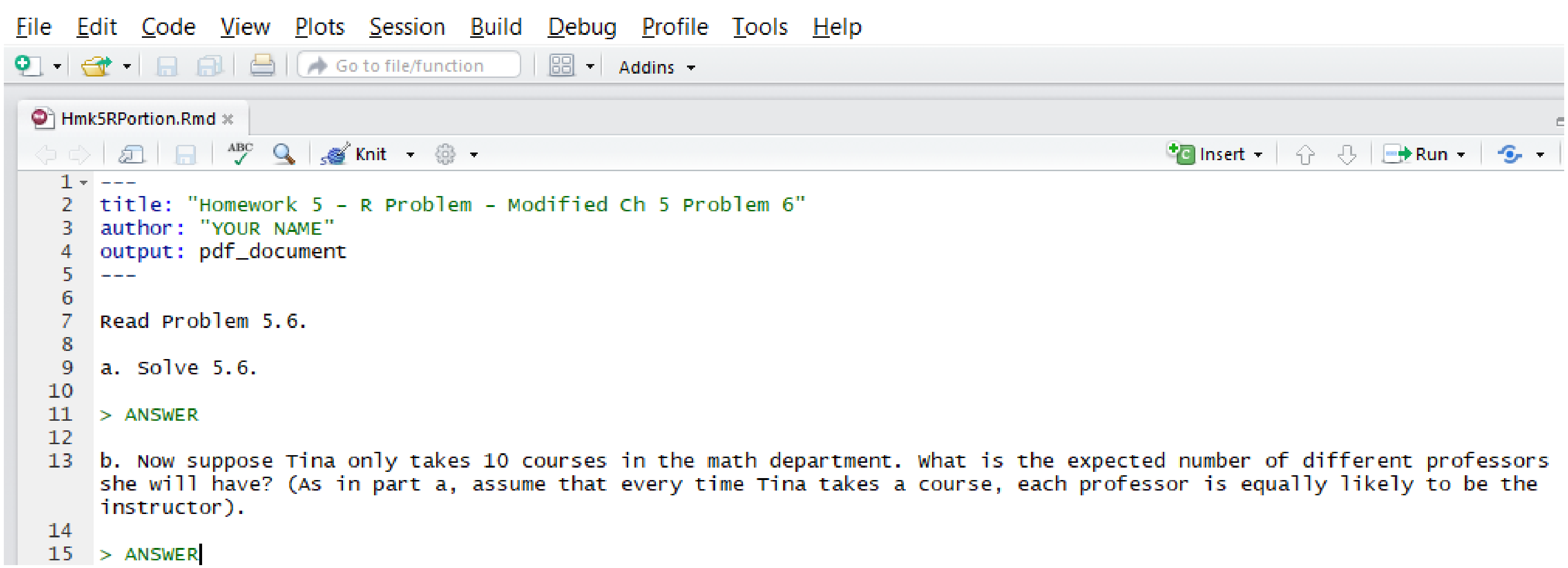
- The first week, students learn:
  - the basic components of a function in R
  - to tweak a function that is provided to create a new function to do a very simple task
  - learn about pseudocode (an outline of appropriate steps)
- The third week, they have to write their own function, and use it to 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.

R Supplements

- Designed to increase student interaction with the software and code
- Took the R code from the textbook and supplemented it with detailed comments about what each step in the code was doing
- Included some additional material not included in the text
- Provided as RMarkdown documents
- Instructed to read them along with the textbook material for each class meeting.
- Provided the students with additional examples to reference when working on their assignments.

Sample Assignment

- An R Problem from a mid-semester homework is presented.
- Students were provided with an RMarkdown template to complete.
- Setting is based off a book homework problem.



Sample Activity

- Part a. Solve Problem 5.6. (Dobrow, 2013, pg. 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.*
- *Ans: 49.77*

Sample Activity

- Part 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. (Ans: 7.48)*

Sample Activity

- Part c. We want to simulate and verify the results in part b. 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.*

Sample Activity

- Part 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.*

Sample Activity

- Part 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

- Used a survey to collect student responses about:
  - R Supplements
  - Scaffolding structure
  - Incorporating writing into the course
- Approved by Amherst College IRB
- All 31 students in Fall 2017 opted in

Survey Results

- Material in the R supplements was useful
- Use of the supplements decreased as the semester progressed
- More engagement with the supplements would be beneficial
- Scaffolding and pseudocode was useful
- Improve feedback on the assignments
  - For more details (exact #'s), see paper or ask!

Results (My perspective)

- Students left the class with stronger computational skills than in previous semesters.
- Students were able to write functions and use them to tackle appropriate problems.
- Writing pseudocode and interpreting code for exams proved challenging for some students.

Conclusion

- Incorporated computational learning goals in Probability
- Can feel challenging; more to teach!
- Refinements to materials and assignments could improve experience
- Met with success; Great benefits for students
- One student's open-ended survey response:
  - "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.
- Dobrow, R. P. (2013). *Probability: with applications and R*. Hoboken, New Jersey: John Wiley & Sons.
- 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.

Contact Info

- Amy Wagaman
- awagaman@amherst.edu

Abstract

- 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.