Computing in the Statistics Curricula: A 10-Year Retrospective

Johanna Hardin and Nicholas J. Horton, with Deborah Nolan and Duncan Temple Lang

The Journal of Statistics and Data Science Education special issue on “Computing in the Statistics and Data Science Curricula” features a set of papers that provide a mosaic of curricular innovations and approaches that embrace computing. As we reviewed the papers we felt that this collection would benefit from the perspective of the authors of the landmark “Computing in the Statistics Curricula” (TAS 2010) paper. We asked Deb and Duncan to take this opportunity to reflect on the landscape when they wrote the paper, to comment on the current situation, and to speculate on the future.

1. (JH/NJH) Your 2010 article called for a broader definition of statistical computing and for greater inclusion of this topic in undergraduate and graduate statistics programs. Could you please summarize your argument? What was your thinking when you wrote the paper?

   (DN/DTL) Ten years ago we thought the gap between statistics education and the need for better computing in the practice of statistics was large and rapidly growing larger. The statistics community needed to shift quickly and dramatically to maintain relevance as a discipline and to better prepare our students for success in modern research and the workforce.

   We recognize that we are not the first to call for statistics education to change to more wholeheartedly embrace computing (see, e.g., Friedman 2001). We think that our main contributions were 2-fold. We aimed to: (1) more broadly define statistical computing and provide concrete suggestions of the kinds of computational tools statistics students should be taught; and (2) raise the teaching of computing from recipes and templates to an approach that focuses on computational reasoning in the context of problem solving with data. We called for a significant expansion of computational training. We asked: What do students need to work with real data, data that are not prettified for a neat demonstration of a statistical methodology? Hand in hand with these tools, we advocated for the statistics community to recognize and embrace the intellectually rich foundation of computational reasoning for data analysis, both applied and academic.

   Our hope was to convince readers that letting students pick up computing skills on their own was putting them at a disadvantage to those more formally trained. We advocated for teaching students computational reasoning, giving them experiences where the ability to reason statistically and computationally can impact their understanding of data and their ability to analyze the data, and we called for providing a strong foundation on which our students can later learn new data technologies on their own. Our goal was to gradually/incrementally push the academic statistics community to: (1) elevate the concepts of computing and computational reasoning for data analysis to the same intellectual level, import, and value as mathematical statistics, (2) indirectly, or by proxy, get students working not just with real data, but with real problems, and (3) prepare students to be full contributors on a team of non-statisticians (and not just rely on others to do the computing, to get the data, etc.).

2. Looking back from today's vantage point, would you have written the same paper? What parts of your paper do you now disagree with? What part do you think is most important?

   Looking back, we do not disagree with any points made 10 years ago. We meant to provide specific computational topics to add to curricula and to promote a pedagogical approach. The broad list of technologies in the Venn diagram in Figure 1 of that paper remains relevant. The topics were loosely arranged in groups and not meant to be exhaustive, but rather indicative of the types of topics that we consider relevant to statistical practice and research. We did make reference to R and to specific R packages, and chose not to include Python as we saw it as similar in nature to the role R played in the underlying intellectual concepts. The diagram is otherwise language-agnostic.

   In terms of pedagogy, we might have changed the focus of the paper to emphasize more the importance of problem solving in real-world settings. The statistics community has called for greater attention to be paid to real-world problem solving, and we advocated that these experiences must include the computational paradigms that are core to that real-world problem solving. For example, in Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving (Nolan and Temple Lang 2015), we aim to expose this problem-solving approach through case studies that illustrate the dynamic and iterative process by which data scientists approach a problem and reason about different ways to implement solutions.

   Much of our work over the past decade has aimed to instill in students a habit of mind for computational thinking. We want
students to learn computing principles, not computer scripting. And, we want them to adopt computational work environments that can help ameliorate the reproducibility crisis in science.

One such habit is to deploy basic coding practices and principles so that code is easy to understand and maintain. For example, the DRY principle, which stands for don’t repeat yourself, is at the root of important qualities of code design, such as abstraction and modularity, and helps us write code that follows the keep it simple (KIS) principle. In other words, if we find ourselves copying and pasting chunks of code while changing only a few aspects each time, we should consider writing a function to do the job more parsimoniously. We identify the common task, create a function to perform this task, and design the changing aspects as inputs to the function. This way, when we need to modify the code, we update it in one place. Our code is more easily maintained and avoids having slightly different versions that are out of sync, contradictory, and error prone. Further, when a function (or any code) becomes too long, has too many purposes, or has too many parameters, we identify smaller tasks and create functions or modules with fewer inputs to carry out these tasks. Ideally, these smaller modules operate without the knowledge of inner workings of the others.

When we clean and prepare data for analysis, we typically carry out several tasks, some of which we have not anticipated, and we often backtrack and iterate parts of the data preparation as we come across new issues with the data in our analysis. As we clean and prepare the data, a good habit is to organize our work into distinct tasks and wrap these tasks into functions that help preprocess the data. This way it’s easier to recognize dependencies, and when we discover a new task, it is simple to repeat part or all of the preprocessing. Other habits of mind address error handling and efficiency. We all make mistakes when we code, and understanding how an error arises is part of computational thinking. If we understand the computational paradigm of a language, then we can write better, more efficient code.

We have found that the best way to instill these habits is to model them for the students where we think out loud as we code, make mistakes, and fix them. These basic programming and software engineering principles underlie the approaches we take in Nolan and Temple Lang (2015).

3. Ten years ago, “data science” had not yet entered the mainstream. How does your advice fit in or not with today’s popularity of this emerging field? What has happened in data science education in the intervening decade?

In 1997, Wu (Chipman and Joseph 2016) called for statistics to be renamed data science and statisticians to be called data scientists, but it was more than a decade before data science really took off. Regardless of the terminology, our thinking has been shaped by the initial “data scientists” familiar to us from Bell Labs—people who could frame a new problem that did not have an off-the-shelf solution, map the problem to data that could be useful to address it, determine how to get those data, analyze them, and make decisions based on the analysis. Traditional statistics education typically involves a short sequence of these steps that all too infrequently involves framing the question and rarely makes a decision based on the statistical results.

We co-developed and separately taught our first course in what would now be called “data science” in spring 2004 at the University of California, Davis and Berkeley. That course was the framework for our 2010 paper. At UC Berkeley, the course, called Concepts in Computing with Data, was required for the statistics major, and it later served as the basis for DATA 100: Principles and Techniques of Data Science, which is required for the new data science major at Berkeley. DATA 100 was first offered in spring 2017 and now enrolls over 1100 students every semester (see http://www.ds100.org for more information).

Today, we find that students have a greater sense of agency and desire to work on meaningful, larger projects than textbook datasets that aim to illustrate a method. We need to both meet this desire and also leverage it to qualitatively change how and what we teach. We need to produce a different type of statistics graduate.

Unfortunately, many who are new to data science are confused about what statistics is as a field. For example, we have talked to students and faculty who do not think machine learning and visualization belong to the field of statistics. Ironically, some of these same people have told us that logistic regression is one of the key tools in the machine learning toolbox. We feel this disconnect keenly. It’s frustrating and disappointing that “statistics” has such negative connotations and is shunned by many students in data science. The introductory Foundations in Data Science course at Berkeley purposely omits “statistics” from its description of its three perspectives: “inferential thinking, computational thinking, and real-world relevance.”

One response to the emergence of data science as a phenomenon was to quickly rename statistics courses and add “data science” to the title while making little change to course contents. Students are not fooled by this renaming. Another approach has been to design a new major in data science that at least at face value appears to be created from a combined list of existing courses in computer science and statistics. Such a major looks nearly identical to a double major in both fields and lacks integration of the principles from these two fields. It is at this interface that students learn data science.

In short, too many in the statistics community are defensive and are stuck in a mindset that does not seem to embrace the new opportunities that data science brings but rather tries to see how to claim overlap with this emerging field by repackaging existing material. There is a lot of activity in “data science” defined broadly as being what they already did. While there is some truth to this claim, we argue this stance is parochial and short-sighted. Fundamentally, the potential for data science is much more than the union or the intersection of Computer Science and Statistics and Math. It would be a shame to miss this opportunity to create the field that Statistics could and should have become a decade or more ago.

Over the past 10 years, we have branched out from statistics to build data science programs and institutes on our respective campuses. Our activities have purposefully aimed at creating an interdisciplinary community and campus-wide collaborations.

Importantly, we think the potential of data science is in the Science of Data Science. Specifically, what is the general framework/theory of developing data-driven questions and solving them? This is something many commonly say can only be attained by experience, much like an apprenticeship. While there is some historical truth to this, we believe that the field can and should develop a framework that is much more efficient,
principled, and accelerated in how we frame, decompose, map, analyze, and answer data-driven questions.

Graduate students can do research in this area of science and philosophy. Masters and undergraduate students can learn from these frameworks how to think about a very broad set of problems at a much richer level to actually deploy (not "apply") them and understand the limitations of the solutions and the opportunities for better solutions. While students learn statistics, computing, mathematics, and other disciplines, the focus of this science is more on the principles of problem framing and solving.

4. What still needs to be done (at the undergrad level? Masters programs? reshaping Ph.D. programs?) Is there one core point that you made 10 years ago that has not yet been addressed?

Much activity in computing over the past decade has gone in a very different direction from the computational reasoning and habits of mind that we have described in this paper. A lot of work has been done to simplify programming in R and Python with a focus on creating interfaces to existing functionality. And there is a greater emphasis on scripting and away from programming. While this approach should be commended for bringing R and Python into the hands of more people, it should not be confused with teaching computational reasoning and fundamental concepts of computing for data analysis.

We are each working on books that aim to bring this reasoning about the R language to the fore in introductory (An Introduction to Programming Principles in R, Fitzgerald, Nolan, and Ulle) and more advanced (The Computing Mindset for R, Espe and Temple Lang) settings.

As for what needs to be done with respect to Ph.D. training, we did not really address research-level work in statistical computing 10 years ago. There was and remains a tremendous dearth in graduate education in statistical computing. To this point, we simply ask: How many Ph.D.s have graduated in the past 10 years with a primary focus on statistical computing? How many have joined the R core team or other computing environments (e.g., Julia)? How many have created software that would be considered a novel research idea and not just implementation of an existing idea? The scarce numbers in the answers point to the problem.

The amount of output in terms of R packages and Python libraries is remarkable given the forces within academia and the statistics profession that encourage focus on other aspects of the field. However, we would not call basic implementation of statistical methods or of learning experiments research output in statistical computing. While these contributions can be useful, they do not necessarily constitute research in the field. We need more innovation in, and higher respect for, statistical computing and need to make it an intellectually rich field if we are to retain and attract top researchers.

5. What do you think of our collection for the special issue? Is it what you would have expected 10 years ago? Is there something missing (either because the profession is not filling in holes or because our collection is incomplete)?

We are honored to have a collection marking the 10-year anniversary of our TAS paper, and we are pleased that there has been so much activity in this field. We are grateful to the editors, Hardin, Horton, and Witmer, for organizing this special issue and to the authors for their contributions.

Rather than comment on how complete or incomplete the collection is, we make two observations about the collection and use these observations as launching pads for our final comments. First, it is wonderful to see so many papers address introductory statistics courses. We need to be involved at these early stages to develop awareness and foundations in students about data science, statistics, and data analysis. Generally, much of the focus in statistics education is on beginners and introductory courses where there is a presumed reluctance, lack of ability, and/or ambivalence about statistical computing. Given the numbers of students in these courses, this is understandable. This balance is founded in realism, service, and expedience. However, we take this observation as an opportunity to urge the statistics community to consider the long-term sustainability of our field. Bringing more immediate focus on advanced education, that is, upper division undergraduate, Master’s and Ph.D. programs, will produce more capable instructors and researchers in the field. We need to train those who will go on to teach the next generation, and train them in the foundations of computational reasoning for data analysis. That’s key to the health of our discipline—well-established and respected faculty who are committed and invested in statistical computing and in integrating it across (nearly) all other statistics courses, not just those who view it as a sideline or who have themselves “picked it up” in an ad hoc manner.

Second, eight of the 18 abstracts from the initial list we saw contained the word “skill,” one five times. While one can read too much into this choice of word, its use does indicate a difference between principles/concepts and less cerebral tasks. The extensive use of the work “skill” reflects the implicit relationship statistics has long had with computing; that is, computing is not an inherent part of the intellectual discipline but something we import and use. Curiously, people in other disciplines that use statistics talk about “statistical skills,” referring to statistics not as an intellectual matter, but a practical, supplementary skill. Statisticians often balk at this description and bring attention to the importance of “statistical thinking” in contrast to statistical recipes. In parallel, we want our students to have more than “computing skills.” We advocate it is high time for the academic statistical community to embrace fundamental and foundational “computational thinking” as an integral part of our field.

Thanks again to the editors and authors for continuing to raise the importance of computing in statistics.

References

Chipman, H. A., and Joseph, V. R. (2016), “A Conversation With Jeff Wu,” Statistical Science, 31, 624–636, DOI: 10.1214/16-STS574.

Friedman, J. (2001), “The Role of Statistics in the Data Revolution?,” International Statistical Review/Revue Internationale De Statistique, 69, 5–10, DOI: 10.2307/1403524.

Nolan, D., and Temple Lang, D. (2010), “Computing in the Statistics Curricula,” The American Statistician, 64(2), 97–107, DOI: 10.1198/tast.2010.09132.

——— (2015), Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving, Boca Raton, FL: CRC Press.