INTEGRATING COMPUTING AND DATA TECHNOLOGIES INTO THE STATISTICS CURRICULA

Deborah Nolan and Duncan Temple Lang
University of California, United States of America
duncan@wald.ucdavis.edu

It is increasingly clear that computing is becoming an essential skill for statisticians and anybody working with data. Computing is as important as mathematics in both statistical practice and research, yet it occupies a tiny portion of our curricula. We have an obligation to reform our upper-division and graduate curricula and integrate computing. We need to change our view of the role of computing in our programs, and teach computational fundamentals and reasoning, rather than ad hoc "tricks" or templates. Furthermore, we must broaden our notion of "statistical computing" to teach modern data technologies. The needs for statistical computing are different from computer science and we must teach this increasingly diverse topic within the statistics curricula. This requires us to fit more into our curricula and also for many of us to learn this material. Computing is important in its own right but can also greatly improve how students learn the traditional material and introduce them to a different aspect of statistics.

THE NEED FOR CHANGE IN THE CURRICULA

We are experiencing immense change in society where data are much more ubiquitous and recognized as being of great importance. As data are increasingly available, evidence-based decision making and an evidence-based society appear to be feasible goals. However, being able to reason about evidence presupposes that we can access the evidence. While the data may be available to us, we need the skills to be able to retrieve them in a manner that makes them useful to us and then to be able to computationally turn them into information and then communicate the results in meaningful ways.

Essentially, the nature of what statisticians do is changing considerably, both in practice and in research, and involves a significant computational component. Unfortunately, we have not seriously adapted the upper-division and graduate statistics curricula. Computing constitutes a tiny proportion of our curricula and is typically taught by teaching assistants (TAs) as needed in order to do assignments rather than as a regular part of the intellectual material of the course. We need to acknowledge and embrace the technological developments of the last two decades. It is important that we do so if our field is to play the role it should in an evidence-based society. As educators, we need to 1) present computing as a serious intellectual topic and teach the students the fundamental concepts of computing and how to reason about computational tasks; 2) broaden the notion of statistical computing to include, and even focus on, modern information and data technologies and not just traditional topics of computational statistics; 3) teach students how to learn about new topics in computation that relate to analyzing data (e.g., visualization, parallel computing, ...); 4) exploit computing to teach modern statistics in new and more interesting ways that actively involve students in statistical practice related to topical, important scientific and social issues. Not only is computing a large part of our work and we need to do it better, changes in data technologies will continue to transform the nature of the work we do and we need to be able to both react to and lead these changes.

We need to seriously evaluate our existing curricula, set aside previous notions and legacy thinking, and approach computing as more than just a tool that changes how we do things, but as a technology that changes what statisticians actually do. Generally, most statisticians think we should have more computing in the curricula. However, the specifics involve many details and trade-offs and even require us to be specific when using general terms like computing. Computing provides a means for broadening not only the reach of statistics but also the nature of statistical engagement. This involves teaching students to understand and think about technology and how it relates to statistics. In short, we need to reevaluate our view of the role of computing in statistics and recognize that not only do students need to learn, but we must also embrace, adapt to and direct changes in technology to improve statistics itself.

In C. Reading (Ed.), *Data and context in statistics education: Towards an evidence-based society. Proceedings of the Eighth International Conference on Teaching Statistics (ICOTS8, July, 2010), Ljubljana, Slovenia*. Voorburg, The Netherlands: International Statistical Institute. www.stat.auckland.ac.nz/~iase/publications.php

Most of our upper-division and graduate students will have taken at most one or two computer classes. They will, however, have taken at least ten years of mathematics classes. While we may lament that their mathematics skills are weaker than we would like, they at least have a mathematical vocabulary and can reason about fundamental mathematical concepts. This is not true for their computing skills. Many of the students have only the most basic knowledge of computing and this severely limits what they can do to access and analyze data. The computer is an obstacle rather than a helpful tool. It is far from clear that when our students graduate that they can do meaningful data analysis. Additionally, they will be involved in data preparation and presentation of results. It is clear that Web services, Web 2.0, interactive and dynamic graphics are going to play a vital role in interpreting data and so we must prepare our students with an adequate background to contribute in these areas in addition to teaching them statistical reasoning and methods.

Research in statistics education has focused mainly on changing the introductory statistics classes. These clearly impact the highest number of students, but not necessarily the students who are most likely to benefit from improved statistics education. We need to focus on the upper-division and graduate programs and modernize these to both improve the educational experience and also attract different types of students. The changing nature of society and data availability should act as a catalyst for us to focus on these aspects of education. While many departments have changed course offerings, there has been relatively little comprehensive change or even discussion of how these should change. While it may be difficult to predict where technological changes will lead, it is clear that computing plays and will continue to play an essential role in both statistical practice and also research. Regardless of the particulars of future developments in computing, we must include computing and data technologies in our curricula to prepare our students for the future.

Many of the important developments in our field predate “modern” computing, and we have a heavy reliance on mathematics to explain and reason about statistical concepts. Mathematics will of course always be important to the fiels. However, it is no longer the only tool students must master. We need to free ourselves from legacy approaches to teaching and reevaluate the skills students need to learn to be able to both apply and research statistical concepts.

WHAT TO TEACH

Computing, or even statistical computing, is a large diverse field. Statisticians use the term to mean quite different things when discussing computing in the curricula. Many think of it as traditional topics such as floating point representation of numbers on a computer, random number generation, efficient and numerically stable algorithms for linear algebra, numerical optimization, and so on. These are all valuable and interesting topics, but their relative importance varies for different categories of students.

Given that most students have taken only one or two computing classes, statistical computing education must focus initially on the fundamentals of programming. Students must learn to efficiently and reliably express computational tasks in the form of instructions to the computer. This involves a programming language and the students must be taught not only the syntax and vocabulary of the language, but how to compose the equivalent of sentences, paragraphs and express computational concepts. They must learn to express themselves clearly and succinctly and to use the programming language to communicate ideas to both the computer and other readers of the code. They must develop a style, not just of formatting code, but of expressing computational tasks and ideas. Students should be able to understand the merits and demerits of different approaches to a problem and be able to appreciate the difference between well-written code and ad hoc solutions. They should also be able to find, evaluate and use existing software.

Students should learn a high-level, statistically focused programming language such as R, S-Plus, Stata or SAS or perhaps MATLAB. High-level languages allow the student to be productive early in the class and focus on concepts and not details. The chosen language should also teach students the common concepts of general programming languages. This will ensure that they learn important long-lasting concepts that transcend specific languages. Accordingly, Stata and SAS are less appropriate for these purposes. Many people argue that we should teach SAS because that is how students will get a job. This argument short changes the role of computing in

modern statistical practice and research. Furthermore, computing and technology are no longer just a useful tool, but something significantly more and not treating it as such illustrates a lack of understanding of the changing nature of statistics.

Courses that teach the foundations of statistical computing and programming need to teach both concepts and specifics. They need to teach the basic data structures, reading data into these data structures, programming constructs such as control flow, writing functions, developing software, and also visualization facilities. They also need to teach the available functions - the vocabulary - and how to do useful things such as create graphical displays of data. Most importantly, computing courses need to focus on abstractions and concepts rather than details and minutiae. The students will have lots of questions about these details, and it is our job to try to teach them how to think about these questions and the general concepts that will allow them to learn these across different languages.

In addition to the fundamentals of programming and programming languages, the students should be taught about the tools that aid computational activities. These include topics such as the tools for and process of debugging, profiling to find bottlenecks, version control, testing of code, and portability. These are important topics for all students, but especially for graduate students. This also illustrates to the students that there is a lot more to learn and encourages the motivated students to pursue these topics.

Given a solid foundation in programming, students need to learn about manipulating data. This includes reading regularly structured tabular data and dealing with missing values. We then move to reading more complex and irregularly structured data. This involves reading parts of a document and filtering or transforming the content. Students use programming constructs and string manipulation. This leads to regular expressions for pattern matching in strings and the important topic of text processing. While this can be done within an environment such as R or MATLAB, we also take the opportunity to introduce the students to the UNIX shell and the benefits of using line-oriented facilities for efficient filtering. A natural progression from text manipulation is accessing data from the Web by “scraping” HTML documents for tables and links or even the text, e.g. comments on blogs. We then move on to more structured Web services which return XML. We build on the same general tools such as XPath that we used for processing HTML documents to extract the content from XML documents. We also introduce the students to relational databases and teach the students how to access data in the database and combine computational tasks from within R and the database. Like Murrell, we also teach aspects of presenting results via the Web. We focus on generating graphics such as interactive Scalable Vector Graphics (SVG, a dialect of XML) and Keyhole Markup Language (KML, another dialect of XML) for display on Google Earth or Google Maps, and embedding these within HTML pages, perhaps with HTML form controls. Many of the details are hidden, but students learn the roles of the essential technologies (including JavaScript) and how they work together.

We have the students work on many of these topics as part of a real problem in data analysis, e.g., classifying SPAM email, predicting location in a building using strength of wireless signal. This helps to illustrate the relevance of the topics and we use the context across several assignments to have the students take a problem from beginning to end. For example, we have them programmatically read email messages using regular expressions to compute derived variables that we then use to train a classifier to predict whether a new message is SPAM or HAM.

There are clearly too many topics to teach in a single course and not all topics are equally important to all students. We think data structures, fundamental programming skills, graphics, data input, regular expressions and Web-based XML/HTML manipulation are the essential topics for upper-division undergraduates. For graduate students, we feel the important topics are data structures, programming skills, graphics, data input, regular expressions, object-oriented programming concepts, creating R packages, efficient computing, the basics of the “UNIX'” shell, interfacing to compiled code such as C/C++ or FORTRAN, and high-level parallel computing. Given the extreme lack of background that most students have in computing, we also feel that one computing class is not sufficient, especially if it is a ten-week quarter-long class. We need to "encourage" students to take more computing classes. We cannot continue to give them the impression, as we do in our current curricula, that computing is trivial and unimportant and something that they can learn on their own or on the side.

HOW TO TEACH

Teaching statistical computing is quite different from more traditional mathematical and methodological topics. For one, students have almost no background in studying computing and typically we only get one class to teach them what they need to know. This makes it challenging to convey both practical information and emphasize the fundamental abstractions and concepts that make the material useful. Secondly, the students do know some computing from interacting with a Windows interface or their iPhone! So we have to work around some of this familiarity as it is very different from statistical and scientific computing.

An important positive difference between computing and traditional statistical topics is that "active learning" is a natural side-effect of the medium in which the students work. Not only do students have to understand the fundamental material, but also actively use it to construct commands that implement the task. This is a significantly different way of interacting with the material than in, for example, mathematical statistics. The students have to build something that gets the right results as opposed to less tangibly, and sometimes more passively, understanding and accepting a theorem without really digesting its significance. Furthermore, the computer offers important feedback that is missing in mathematics. Students issue commands and get results or errors. They can learn from these responses and tune their commands to get the desired result. This active construction and interactive feedback makes computing a very different medium for learning. While frustrating and time consuming to learn, it is a very rich learning environment that we can exploit not just for teaching statistical computing, but statistics concepts in much more hands-on and meaningful ways than we typically have.

As we have mentioned, it is essential to strive to teach the abstract ideas and concepts of computing to students in a statistical computing class. It is easier and more immediately rewarding to show them how to do specific tasks. These are short-term lessons. We must continue to focus students on the abstractions that will continue to be important even as technologies change.

Because of the students' lack of background in computing, we must explicitly cover the details of new concepts in addition to providing the big picture view of the topic(s). However, since the material is quite new in both content and nature, we can also teach in new ways. For example, rather than simply lecturing on material, one of us (Temple Lang) starts each class by asking for questions. Students are given assignments that require them to work at a regular pace over two weeks. As they do so, they encounter problems and are urged to raise these in the classroom. This allows the abstraction of the question and connection to other important concepts. Most importantly, it allows us to go through the thought process to answer the question, guiding a discussion about how the student might be able to answer the question him or herself in the future. This thought process is not something students can easily learn from a book, unlike the details of the syntax and function calls. We need more "discussion-like" classes in statistics not just in computing but on data analysis rather than teaching the procedural material that is in so many books.

We give assignments that the students typically work on for two weeks. They are reasonably comprehensive tasks that illustrate reasonably realistic activities in modern data analysis rather than being artificial "baby" problems. When the students start, they see the high-level sequence of sub-tasks for the whole process but have not necessarily seen the material for solving each step. The students are exposed to the process by which we actually approach data analysis problems, separating the steps and deferring details about how to do a particular task until later or finding alternative temporary approaches.

We make extensive use of mailing lists. Students post questions about difficulties they are having or new insights they have discovered. Other students, along with the instructor and TA, respond to questions. The act of writing a question is an important part of the learning experience as it requires the student to formulate the question clearly and to distill the problem they are experiencing into something they can articulate to others. The students also gain experience in leveraging mailing lists that will serve as valuable resources for the future. Part of the student's grade comes from their participation, including their activity on the mailing list, either answering or asking questions.

We actively encourage our students to find and use material from the Web and from other students, as long as it is properly acknowledged. This teaches the students important skills that they will use in their career.

As with applied statistics, it is hard to give regular written exams in statistical computing courses. Students work in the interactive medium of the computer throughout the course, and it does not make sense to test them using a different medium. We believe the students learn from the active engagement their assignments involve. Therefore, the exam is less essential to consolidate the information throughout the course. We do attempt to make the final assignment or project sufficiently inclusive of many of the topics so that a) the students have to have mastered the important elements of the course, and b) that they experience a reasonably complete data analysis project that they might experience in the future and carry it through from beginning to end, determining which tools to use. Ideally, the final project (along with the other assignments) could be used by the student as a portfolio when applying for graduate school or jobs.

Given that a statistical computing class is quite different from any a student has taken before (even from computer science classes), students find our classes somewhat unusual in form, very time consuming and challenging in content. However, the majority of the students find it creative and a very important part of their learning that leads to success in jobs and research.

Aside from the need for more computing in the curriculum, we feel that the way regular statistics topics are taught needs serious overhaul. Statistics classes tend to focus on a sequence of methods and claim to be "applied" by illustrating these methods with particular data sets. We feel that this is quite unlike statistical practice in which an analyst must try to frame and answer questions and select statistical methods to help. As a result, the students do not learn how to function as a practicing statistician through regular classes. We attempt to use our computing classes to address this experience. We give students real and topical data and have them explore it. We give them some possible questions to explore but actively encourage them to pose their own and investigate those and even find additional data. Often we give them data that constitute the entire population (e.g., baseball performance information) and so there is no sampling variability. This focuses the students on interpretation rather than mindlessly putting numbers through statistical formulae.

We also use our computing class to introduce modern statistical methods that the students may not see in traditional classes. For example, we might show them classification techniques such as k-nearest-neighbors, recursive partition trees, naive Bayes, boosting, random forests, Markov chain Monte Carlo (MCMC). In addition to seeing new statistical techniques, the students also experience exploring and learning about new software through help pages. This helps to teach them how to learn about new software and technologies.

For various reasons, we might be tempted to delegate computing to computer science classes. This would be a serious mistake. While computer science and statistical computing have significant elements in common, the nature of statistical computing is very different from what is taught in computer science classes. Statistical computing is about using the computer to make sense of data. It is inherently interactive, with the next step determined dynamically by the content we see in the previous steps. It is the tightly coupled combination of programming, visualization and statistical reasoning. Even debugging is empirical. Computer science programming classes are about programming language constructs, developing complete, structured software that can be specified before the data are collected. Furthermore, more advanced classes in computer science such as on database management systems provide too much detail relative to what most statisticians need to know, i.e. how and when to use databases for analyzing data.

ACTIVITIES TO FOSTER STATISTICAL COMPUTING EDUCATION

We have described the need for fundamental change in our curricula to integrate computing. This should involve dedicated classes on computing topics. It should also involve more computing in existing classes that teach computing in context and give the students a different medium with which to explore and understand the concepts being taught. While proposing such changes, we also need a plan of action to implement these.

One reason why we in statistics have been slow to integrate computing into the curricula is that few of us have much training in computing. This quite understandably limits how well we can

teach the topic. Therefore, we need to teach instructors how to teach statistical computing classes and topics. This covers explicit computing classes and also integrating computing into existing statistics classes. Many instructors also need to be taught the material itself in addition to how to teach the topics. We have organized two workshops funded by the National Science Foundation in the US. to this end (Nolan et al., 2007). We think that we need to have several more over the next few years until we have sufficient number of graduating PhDs that will have learned this material in their coursework.

Teaching materials such as a books, sample syllabi, assignments and case studies, lecture notes, etc. are also extremely important in facilitating instructors introducing computing topics into the curriculum. Without these, the obstacles to get started may be too much for some instructors. Books on these topics are starting to be published, e.g., Murrell (2009). We are authoring one on programming and data technologies aimed at the upper-division and introductory graduate level (Nolan & Temple Lang, 2011). We have also made our course materials and assignments available from our web sites (*www.stat.ucdavis.edu/~duncan/* and *www.stat.berkeley.edu/~nolan*).

We are also in the process of editing a book of case studies in computing with data. The book is a collection of approximately 15 case studies, each authored by one or more statisticians, that illustrate an interesting topical problem and how to access and manipulate data to address the associated questions. The case studies are being written as interactive documents and to be accessible via Web browsers to allow students and instructors to explore different aspects of the problems and see the process of reasoning about the computations and questions.

We have published annotated syllabi for various different computing classes (Nolan et al. 2007, Nolan & Temple Lang, 2010).

CONCLUSION

Computing is an essential part of statistics and a pitiful part of statistics education, both in quantity and intellectual content. We need to change our curricula to integrate computing. This needs to happen by introducing explicit courses on computing that teach the fundamentals, vocabulary and computational reasoning and also information and data technologies. But we also need to explicitly include computing as part of existing courses and treat the computational aspect as an important element and as an opportunity to teach differently and better.

We need to decide how to embrace technological changes and allow them to help us redefine the nature of statistical practice and research. We can effect major changes in statistics and wider by changing our curricula to adapt to and adopt ever-evolving modern technologies, positioning ourselves and our students as leaders in this domain. We can teach more interesting statistics in context using these technologies and revitalize our programs. We can attract good students. All of this will involve a lot of work, but it is the right thing to do. If we don't do it, other disciplines will justifiably fill the void.

REFERENCES

Murrell, P. (2009). *Introduction to Data Technologies*. Boca Raton: Chapman & Hall/CRC.

Nolan, D., & Temple Lang, D. (2009). Approaches to Broadening the Statistics Curricula. In M.C. Shelley, L.D. Yore & B. B. Hand (Eds.), *Quality Research in Literacy and Science Education*.

Nolan, D., & Temple Lang, D. (2010). Computing in the Statistics Curricula. *The American Statistician*, to appear in *64*(2).

Nolan, D., & Temple Lang, D. (2011) *Scientific Computing with Data*. New York: Springer. In preparation.

Nolan, D., Temple Lang, D., & Hansen, M. (2007). Computing in Statistics: Model courses and curricula. *www.stat.berkeley.edu/users/statcur*/.