Teaching Quantitative Methods: What Makes It Hard (in Literary Studies)

Citation to Publisher Version: Goldstone, Andrew. (2018). Teaching Quantitative Methods: What Makes It Hard (in Literary Studies). In Matthew K. Gold, Lauren F. Klein (Eds.), *Debates in the Digital Humanities 2018* (-). Minneapolis: University of Minnesota Press.

Citation to *this* Version: Goldstone, Andrew. (2018). Teaching Quantitative Methods: What Makes It Hard (in Literary Studies). In Matthew K. Gold, Lauren F. Klein (Eds.), *Debates in the Digital Humanities 2018* (-). Minneapolis: University of Minnesota Press. Retrieved from http://dx.doi.org/doi:10.7282/T3G44SKG

ANDREW GOLDSTONE

Teaching Quantitative Methods: What Makes It Hard (in Literary Studies)

Forthcoming in *Debates in the Digital Humanities 2018* (University of Minnesota Press).

When I set out to teach a graduate English course titled "Literary Data: Some Approaches" in the spring of 2015, I was on a mission.[1] I wanted my eleven Ph.D. students to learn not simply how to talk about DH but how to analyze data as part of their literary scholarship, to be able not only to argue about "data" but to argue *with* data. I wanted to prove that English graduate students could do more than play with computers in their first DH course. At the same time, I wanted students to acquire the conceptual sophistication that would make their practical knowledge meaningful. Though my students made remarkable practical and conceptual progress, at the end of the semester my high-flown aims still seemed to lie beyond our immediate grasp. Having made the attempt, however, I learned some lessons of my own about what is needed in order to take on this pedagogical mission: not only lessons in course-planning but lessons about what the scholarly community has to do—and what institutions must be prepared to supply—if quantitative methods are to fulfill their promise for the study of literature and culture.

There is, of course, more to DH than doing data analysis for literary study. Aside from a disciplinary slant towards my home field of literature in English, my course's emphasis on data analysis meant giving relatively less attention to other prominent strands of DH, for example new media studies or digital editing and curation. Nonetheless, questions of what constitutes literary data and how it can be analyzed are central to DH discussions. DH has garnered so much attention and debate across humanistic scholarship in no small part because of the claim that studying aggregates of texts with the help of quantification might radically change our conception of literary and cultural history. The potential of aggregation is the reason I have participated in DH. And if it really does open up major new possibilities for research, then teaching the theory and method of literary data analysis to future researchers is something to which at least some literary scholars ought to devote serious energy.

My argument, in brief, is that teaching this material is really, really hard, for reasons that are more than technical or technological. The available strategies for teaching literary data analysis under the "DH" rubric, including my own, have so far been inadequate to the task of training scholars in research methods. In what follows, I'll summarize how

1. The course syllabus, with a bibliography of readings, is available online from the MLA CORE repository at http://dx.doi.org/10.17613/M69S30; the course assignments (ten problem sets and two papers) are available at http://dx.doi.org/10.17613/M6602R. For their thoughtful comments on earlier versions of this essay, I thank Taylor Arnold, Natalia Cecire, Anne DeWitt, Meredith McGill, David Roh, the *Debates in DH* editors, and several students from English 350:509, Spring 2015. I am grateful to all my students for teaching me much of what is discussed here.

I approached this pedagogical challenge in my course. But rather than offer my course as a model, I instead draw the following prescriptive lessons from my experience:

1. Cultivating technical facility with computer tools—including programming languages—should receive less attention than methodologies for analyzing quantitative or aggregative evidence. Despite the widespread DH interest in the former, it has little scholarly use without the latter.

2. Studying method requires pedagogically suitable material for study, but good teaching datasets do not exist. It will require communal effort to create them on the basis of existing research.

3. Following the "theory" model, DH has typically been inserted into curricula as a single-semester course. Yet as a training in method, the analysis of aggregate data will undoubtedly require more time, and a different rationale, than that offered by what Gerald Graff calls "the field-coverage principle" in the curriculum.[2]

Nor are these issues only pedagogical. I learned with my students that present-day literary scholarship is fundamentally uncertain about how to make a convincing scholarly argument using quantitative evidence, and it has a habit of postponing or avoiding methodological debate in the name of open-ended exploration or disciplinary pluralism.[3] If a more sustained research program including quantitative methods is to develop, those dispositions may be reaching the end of their usefulness.

TEACHING "LITERARY DATA"

Since I aimed for my students to learn practical skill and conceptual sophistication, I decided to divide each week's three-hour meeting in two: half the time for seminar dis-

2. "The division of fields according to the least controversial principles made the department easy to administer but masked its most interesting conflicts and connections" (8). We might add that it makes any kind of cumulative training difficult to formalize.

3. Most typical is the appeal to the complementarity of "close" and "distant" readings, as in Hoyt Long and Richard So's "method of literary pattern recognition that is enriched by points of confluence between multiple ontological scales of interpretation" (262). Steven Ramsay argues that "algorithmic criticism" should appeal to a literary-critical audience that is "less concerned with the fitness of method and the determination of interpretative boundaries…[than with] evaluating the robustness of the discussion" (*Reading*, 17). Matthew Kirschenbaum challenges antagonistic accounts of DH in general: "Digital humanists don't want to extinguish reading and theory and interpretation and cultural criticism. Digital humanists want to do their work" (56). Kirschenbaum's statement is a salutary corrective to caricatures of DH, but it also tends to defuse genuine conflicts about how to produce and interpret evidence in scholarship along the lines of field-division discussed by Graff.

cussion of theoretical readings, and half the time for practicum.[4] The major argument of the theory part was that the question of literary data is not restricted either to DH or to its most obvious predecessor, humanities computing, and should be set in a longer and broader multi-disciplinary trajectory. For example, we read Lévi-Strauss's structuralist theory of myth as a transformation of narratives into data; everyone was immediately interested in his system of index cards for tracking mythemes.[5] We looked at McKenzie on scientific bibliography in "Printers of the Mind"; we had a lively discussion of one of Robert Darnton's earliest essays on the history of the book. Turning more explicitly to literary study, our discussion of J.F. Burrows's *Computation into Criticism* was particularly productive, not so much because Burrows's version of computational text analysis struck any of us as compelling but because it presented a limit-case my students and I came back to over and over again: Burrows, fully committed to criticism in the sense of the exercise of judgment, claims that the quantitative patterns of word-use he finds reveal Jane Austen's authorial "genius."[6] But we were also able to see Burrows, and the stylometric tradition he builds on, as only one among many possible models for a literary data analysis.

Our attention then turned to more contemporary work in DH, with short side excursions into contemporary sociology of culture. In our discussions of recent work, two themes recurred: first, my students' excitement and enthusiasm for the novel possibilities opened up in recent work in studying large-scale aggregates of texts; second, their dissatisfaction with the way this scholarship has analyzed and interpreted its data. More than once, my students wished they could study the data being discussed in our readings themselves. They were looking not just for provocative claims—which we found and argued about in abundance—but for models of research they could themselves build on. I will return to this fundamental point later.

The other half of the course was the practicum. My students were to learn R well enough to be able to prepare real-world, messy data for analysis, and to describe it numerically and visually in basic but interesting ways. But I also wanted them to *think*

4. I found a few syllabus models to draw on, especially Matthew Wilkens's Spring 2014 graduate course at Notre Dame (http://mattwilkens.com/teaching/digital-humanities-graduate-seminar-spring-2014/). Matt is among the many to whom I owe thanks for discussions and inspiring examples; see the acknowledgments on my syllabus. To those acknowledgments I add my thanks to Francesca Giannetti, Digital Humanities Librarian at Rutgers—New Brunswick, who visited the class for a crucial session on finding good sources of literary data.

5. I got the idea for using Lévi-Strauss's "Structural Study of Myth" in this context from a blog post by Nick Seaver: "Structuralism: Thinking with Computers," Savage Minds (blog), May 21, 2014. http://savageminds.org/2014/05/21/structuralism-thinking-with-computers/.

6. "Its real force," he says of a principal components analysis of word frequencies in the free indirect discourse of Austen's protagonists, "is as an illustration of the capacities of genius" (175).

programmatically, or, as I put it under "learning goals" on the syllabus, "understand the fundamentals of computation." I know only one general way to teach this sort of skill, and that is through cumulative practice in which skills are used, tested, and reinforced. From this followed the improbable scenario of me devising, and English Ph.D. students completing, weekly problem sets, following, more or less, the progression of a typical basic introductory course in computation.[7] I adopted Jockers's *Text Analysis with R for Students of Literature* as a starting point, but used only parts of the book and added many exercises of my own. Fairly soon, as the challenges built up, I found myself needing to take class time to go over homework questions; since I was also presenting new material in the practicum, this was frequently an overloaded half of the class. The seminar discussion felt positively relaxed by comparison.

All this led up to the final challenge of the course, a data analysis report.[8] Halfway through the term, my students formed pairs (and one triple). Then, over the succeeding weeks, I worked with them as they obtained a data set, analyzed it, and used their findings to produce argumentative papers with code and data appendices. Each student wrote their own report on the basis of their group's work. The folder containing these reports occupies more than seven gigabytes on my hard drive, representing analyses of regionalism in American newspapers; of deaths and absences in the eighteenth-century slave trade; of authorship in science-fiction pulps; of the reception of David Mitchell's *Cloud Atlas*; and of the influences of and on Freud in turn-of-the-century writing. As this list suggests, my students' interests were quite varied, and I encouraged them to apply quantitative methods to whatever topic interested them.[9]

The rationale for the assignment was to confront the class with making sense of data in the wild, with all the messiness that entails. My students' reports bear witness that they all learned that getting the data into analyzable form—"cleaning" it—is a large proportion of the work of analysis. Here, their efforts at mastering R really bore fruit, allowing them to wrestle big bodies of data into sense. (There were more than a few computer crashes on the way, and one student's hard drive filled to capacity in the course of corpus-processing.) Everyone found interesting phenomena to comment on in their data, and

7. In particular, I transposed the outline of John Guttag's *Introduction to Computation and Programming Using Python*.

8. *Not*, I had decided by halfway through the semester, a "project," a ubiquitous DH term which nonetheless suits engineering enterprises, and what Boltanski and Chiapello call the "projective city" of contemporary management theory, better than scholarly arguments (105).

9. On the other hand, my early-modernist students had a much harder time finding materials than my nineteenth- and twentieth-centuryists. By the same token, the project on David Mitchell had to work around the impossibility of obtaining a corpus of twenty-first century novels, turning to a collection of reviews instead.

everyone put particular effort into visualizing these phenomena thoughtfully.[10]

But, after an exhaustingly difficult semester for both my students and me, what of the hoped-for payoff: the promise of moving beyond experimenting with the computer and towards making quantitatively-evidenced arguments? On the theoretical plane, a very healthy caution about drawing conclusions from the data prevailed, even as the class strove to respond to my demand that they make literary-historical arguments. I was gratified at the thoughtfulness—and the quite awesome amount of labor—represented in the reports. At the same time, it was clear that my students did not have all the methodological tools they needed to draw the conclusions they wanted. Most of the projects, for example, involved spotting trends over time: trends in word usage, in network clustering, or in some other phenomenon. But how do we know a trend is real, and not a random fluctuation? Many also sought to explain contrasts among subsets of their data. But how is this to be done, how far can the data allow us to assess the validity of such explanations? Other students needed more possibilities for choosing data to collect in the first place, but how were they to know which choices had a chance of yielding meaningful conclusions?

These questions—the classic questions of quantitative methodology—remained open in the final reports. But this is no criticism of my students, who, as a group, did all I could have asked. Rather, it is a criticism of my pedagogy, which did not equip students with all they needed. Though no doubt my personal shortcomings and all the hiccups of a first-time course contributed to this problem, I think solving it is not simply a matter of tweaking the syllabus or assignments. Instead, it requires a reorientation of the course towards the aim of *teaching methodology*, by which I mean the formal study of the questions I have just posed about how to interpret data. But to do that raises the fundamental pedagogical problems I mentioned at the start.

PROBLEM 1: PROGRAMMING IS NOT AN END IN ITSELF

An informal consensus seems to have emerged that if students in the humanities are going to make use of quantitative methods, they should probably first *learn to program*. Introductions to this dimension of the field are organized around programming languages: *The Programming Historian* is built around an introduction to Python; Matthew Jockers's *Text Analysis with R* is at its heart a tutorial in the R language; Taylor Arnold and Lauren Tilton's *Humanities Data in R* begins with chapters on the language; Folgert Karsdorp's *Python Programming for the Humanities* is a course in the language with

10. Lauren Klein's essay on visualization and archival absence, the subject of a vigorous discussion in the seminar, also made a notable mark on the reports.

examples from stylometry and information retrieval.[11] "On the basis of programming," writes Moretti in "Literature, Measured," a recent retrospective on the work of his Literary Lab, "much more becomes possible" (2).

Steven Ramsay makes the case for focusing on programming in digital humanities teaching. In a 2012 essay, he describes a course that "proceeds, as most courses in programming do, through the major constructs and concepts germane to any programming language" but that nonetheless has a specific rationale for humanists: "the particular form of engagement that characterizes the act of building tools, models, frameworks and representations for the traditional objects of humanistic study" ("Programming"). Here is the kernel of a broad justification for the study of programming in humanities fields, which I found appealing when I first planned my course: learning to program promises to open the door to sophisticated thinking about the scholarly uses of data in our disciplines. At my most idealistic, I even imagined that programming, by making abstractions more concrete, would reintroduce my humanist students to the beauties of mathematics.

The conceptual reward, furthermore, is in principle coupled to a practical one: programming means not being limited to the possibilities offered by any given all-in-one application or "tool." Much DH instruction has foregone programming in favor of guiding students in the use of particular specialized programs.[12] A student who learns to operate the text-analysis web application Voyant Tools, for example, can generate its (long) list of tabulations and visualizations of a text corpus, but no others: the list of visualizations, and of possible dimensions of variation, is closed to non-programmers. Additionally, Voyant makes a significant body of choices about how to convert the text into units of analysis, with no possibility of alternatives beyond the choice of stop words.[13] And if the project goes the way of most DH projects and stops being updated or maintained, what are students who have depended on it to do? By contrast, a student who learns to program in an established programming language has the ability to generate any tabulation or visualization of texts that she can dream of; she can rely on the relative longevity of

11. A rather different focus (with more emphasis on a broad range of software tools rather than a programming curriculum) is found in Shawn Graham et al.'s *Exploring Big Historical Data*. This book, like Arnold and Tilton's, was not available in time for my course. None of these texts is limited to programming, but they all tie quantitative methods tightly to programming, as Moretti does.

12. One example of this approach can be found in Alan Liu's Fall 2014 syllabus for a graduate introduction to DH at UCSB: http://eng236introdh2014f.pbworks.com. The course offers practicums in each week, but turns to new tools for each task.

13. The list of tools in Voyant is found at http://voyant-tools.org/docs. The program was updated to version 2.0 as this essay was being revised. Voyant is, to be sure, a particularly extreme exemplification of the "tool" tendency in DH, but the general phenomenon is widespread; in the *Programming Historian*, aside from the sequence of lessons on Python, one finds numerous lessons which focus on recipes for operating particular programs (Antconc, MALLET, Palladio, etc.).

the platform; and, most importantly, she has the means to transform her texts and other data as required to make her subsequent analysis meaningful.

In principle. In practice—and I had to teach programming to understand this—the distinction between the tool user and the heroically flexible programmer is not so clear. Beginning programmers typically begin by following recipes, modifying them only little by little as they learn what can and can't be done in the language. The first thrill of programming, *making the computer do something*, is a tool-user's thrill. And beginning data analysts cannot know the range of possible methods they might use. They know what they have studied, and, especially in the early stages, they have to spend time simply learning how to apply the techniques they have been shown, by mastering whatever particular steps are needed to use the technique in the programming language they are learning. It's no accident the tutorial form is so ubiquitous in DH instruction. Jockers's *Text Analysis*, chap. 3, for example, is a tutorial in finding the most frequent words in *Moby-Dick*: the whole of the chapter is devoted to explaining the particular R steps necessary to go from the digitized text to the table (and graph) of words and frequencies. This is followed by exercises in finding the top ten words in *Sense and Sensibility*. What students are meant to learn is just precisely *how to extract a table of top ten words and their frequencies*. When I added my own exercises to this chapter, I found myself again asking students to make lists of "top words" in other texts. And this same process of extracting "top words" reappeared in my students' final projects.[14] The infinite generality of programming does not survive classroom necessity.

Yet the problem is not that DH textbooks—or my own course—do not instill the ability to write sophisticated, original programs. In fact, judging from my students' program code in their final analysis projects, either because or in spite of my methods, my students made good progress in acquiring this ability. Anyone who completes Jockers's textbook, Karsdorp's lessons, or Arnold and Tilton's book will have made strides as a programmer, and will know, in a hands-on way, the elementary principles of both algorithms and data structures. But programming competence is *not* competence in analytical methods. It might allow students to prepare data for some future analysis, and to produce visual, tabular, numerical, or even interactive summaries; *Humanities Data in R* gives a fuller survey of the possibilities of exploratory data analysis than the other

14. I was ultimately dissatisfied with the exposition, the worked examples, and the exercises in *Text Analysis with R*; it is written for a different kind of course than the one I was teaching. Nonetheless, literary studies owes a considerable debt to Matt Jockers for his textbook, which begins the serious conversation about what a course in literary data analysis ought to look like, including what sorts of assignments and problems are useful and tractable. As my course went on, I found myself needing to supplement it more and more. But having something to supplement was absolutely indispensable, and the considerable effort of providing additional material taught me how difficult producing a textbook in this domain must be.

texts.[15] Yet students who have focused on programming will have to rely on their intuition when it comes to interpreting exploratory results. Intuition gives only a weak basis for arguing about whether apparent trends, groupings, or principles of variation are supported by the data. Without any sense of these possibilities, which are the stock-in-trade of statistical methodology in the social sciences, a programming curriculum can bring students—but really I should say, it has brought DH as a field—only up to the threshold of method and not over it.

PROBLEM 2: ANY OLD DATA WILL NOT DO

In the words of Miriam Posner, "it's just awful trying to find a humanities dataset."[16] Until I taught my course, I did not anticipate just how hard it would be to find suitable datasets for study. I made the Internet-age mistake of thinking all sorts of interesting things must already be out there for the downloading; and failing that, I imagined, I would be able to construct interesting data sets out of big grab-bag text corpora: Project Gutenberg, the text section of the Internet Archive, and the Text Creation Partnership releases from Eighteenth-Century Collections Online (ECCO) and Early English Books Online (EEBO). I was thinking like an English professor: when I teach literature, I can normally choose a dozen or so interesting books and wait for students to find their own analyses. Students never fail, in the course of reading and discussion, to discover both challenging questions and meaningful evidence to address those questions. But an arbitrary corpus of texts is a different kind of challenge, because *many possible corpora are entirely unsuited to answering any interesting questions*.

Consider again Jockers's textbook. His first corpus is simply the body chapters of *Moby-Dick* (taken from a Project Gutenberg plain text). *Text Analysis with R* invites students to study how the words "whale" and "Ahab" occur with varying frequency across the chapters of the novel, and to assess the correlation between them; later, students can also study the words that occur in proximity to "dog." As an introduction to text-processing functionality in R, this works fine. But these exercises lead to nothing resem-

15. Exploratory data analysis has a methodology of its own, for which the founding text is John Tukey's book of that name. In their chapter in this volume, Arnold and Tilton argue for the central importance of exploratory analysis to humanistic data analysis; they do not identify analytical method with pure programming know-how. Yet scholars who confront bodies of data without any knowledge of what Tukey called "confirmatory" analysis will be limited in the arguments they can make. What is worse, they will have no disciplined way of judging the validity of arguments relying on quantitative evidence. Tukey himself insisted that the two analytical modes "can—and should—proceed side by side" (vii). Qualitative arguments in literary studies are rarely limited to exploration; it is hard to see why quantitative arguments should be.

16. "Humanities Data: A Necessary Contradiction," Miriam Posner's Blog, June 25, 2015. http://miriamposner.com/blog/humanities-data-a-necessary-contradiction.

bling a scholarly argument: what does knowing the changing counts of these words allow us to say about the novel?[17] Later, Jockers shows how to use hierarchical clustering on frequent words to discriminate authorship in a small corpus of Irish-American novels. This is an impressive technical feat, but authorship attribution is rarely an important question in literary studies, and the field still awaits a demonstration that stylometry speaks to central research questions.

The problem is not really Jockers's particular choices. Arnold and Tilton carry out a technically impressive demonstration analysis of a collection of Conan Doyle's Sherlock Holmes stories using natural language processing. But the most elaborate conclusion they reach is a visualization of where main characters occur within each story. Again, the effort is out of all proportion to the result, which any graduate student in English would immediately recognize as at best a *further fact to be explained* of at most moderate interest. In fact, most of the interesting questions about plot are not answerable within a single-author corpus. My own variation on the theme was no better: I spent a whole class carrying students through an analysis that showed the differentation of fiction from poetry in terms of common-word frequencies in the modernist little magazine *The Egoist*. By crude measures, the diction of poetry differs from that of fiction in this publication. So what?

All of these examples have two things in common. First, they work with convenience corpora: collections chosen for practice rather than having been designed to answer research questions. Second, though these collections encapsulate a certain amount of textual variation, they lack interesting metadata. But answering research questions about the relations between text and metadata is precisely what students really need to practice. Students must work with bodies of data that hold meaningful patterns, divergences, and historical developments, and they must learn the techniques necessary to bring those patterns compellingly to light.

Thus, the best data for teaching are not to be found by taking *all* of the bulk releases from big digital archives. Rather, what students need—how unhumanistic this will sound—is data about which at least some answers have already been given, so that instead of being forced to fish for interesting phenomena in an empty ocean, students can follow a trajectory from exploration to valid argument. Such datasets might be designed

17. At this point one might think of the "deformance" argument given well-known form by Jerome McGann in *Radiant Textuality*: any transformation of the text is potentially a provocation to further interpretation. Ramsay's *Reading Machines* applies this thesis to quantitative textual analysis. (We discussed both these texts in the seminar.) By conveniently equating scholarship with creative performance, this argument liberates scholars from responsibility to their evidence and to their community, asking instead only for post hoc rationalization. Research can leave room for many forms of experimentation, but wide use of the "deformance" argument has had a cost: it has forestalled debate about how to use quantitative or aggregate textual evidence systematically.

for teaching by a scholarly collective that sifts freely-available materials and chooses samples to represent an interesting range of variation.[18] But the richest source for such data is research itself. Limiting myself to works from the seminar part of my syllabus, examples might include: the collection of novel titles (with dates and parts of speech) in Moretti's "Style, Inc."; a list of persons mentioned in each of the letters of Thomas Jefferson, as used by Klein to study the collocation network of James Hemings in her "Image of Absence"; the bibliographic metadata on Irish-American novels anayzed by Jockers in *Macroanalysis*; the catalogue of geolocated place names found in nineteenth-century American novels by Wilkens in "The Geographic Imagination." I am imagining that researchers like these might consider making their datasets, already cleaned and enriched, available for student use, within whatever restrictions of copyright are necessary.[19]

One rationale for my course's final assignment, I said above, was the importance of learning to deal with data in the wild. But I now think this emphasis was misplaced. Though data-wrangling is indeed a crucial research skill, it should not come before the question of how to analyze data that *are* in analyzable form: otherwise, why wrangle? I thought I could combine wrangling with more complex forms of analysis, but it is simply too much to fit into a single course, and treating wrangling as an end in itself means deferring past the end of semester the kinds of analysis that serve scholarly arguments. Better to start from tamed data—just as is standard in teaching quantitative methods across all the social sciences—in order to focus on methods for answering substantive questions.

PROBLEM 3: THE SINGULAR DH COURSE IS A BAD FIT FOR THE SUBJECT

Data-wrangling and programming skill often (and, I am suggesting, necessarily) competed with analytical method for time on the syllabus. Still, a more fruitful relationship sometimes developed between the theoretical readings and the practicalities of analysis. Starting about half-way into the course, it became possible to discuss theory and methodology in other scholars' work from the standpoint of practitioners. Once I introduced the grammar of graphics via Wickham's ggplot2 R package, we were able to

---

18. Alan Liu has begun the work in his listing (and production) of Demo Corpora, but whether the already-available corpora support interesting investigations remains to be seen. Liu, "DH Toychest: Digital Humanities Resources for Project Building," http://dhresourcesforprojectbuilding.pbworks.com.

19. It is very hard to point to a finished piece of literary scholarship for which the source dataset is available in a form that would allow students to retrace the analysis straightforwardly. Ted Underwood and Jordan Sellers's work on the reception of nineteenth-century poetry, which they circulated with a "replication archive" after I had finished the course, is a rare example. See Underwood and Sellers, "How Quickly Do Literary Standards Change?," preprint, May 19, 2015, doi:10.6084/m9.figshare.1418394. and, for my own blog post about re-using this archive, "Of Literary Standards and Logistic Regression: A Reproduction", January 4, 2016, http://andrewgoldstone.com/blog/2016/01/04/standards/.

have a very lively discussion of the construction and implications of specific visualizations in work by Lev Manovich and Lauren Klein.[20] Or again, after a lesson on defining and using functions, we were able to be much more concrete in our debate about Tara McPherson's use of the concept of "modularity" in her important essay on race in DH. And after tedious weeks of counting single words in texts, the rationale for topic modeling was particularly vivid.

Yet each of these exhilarating moments depended on the previous weeks of practicum preparation, and as soon as each session ended, it was time to turn to a new topic, with its own challenges. And because of my decision to devote so much time to programming, data analysis itself remained mostly obscure. For example, in the course of the semester I presented three ways of exploring variability in data: correlations between word frequencies (from Jockers); Latent Semantic Indexing; and Latent Dirichlet Allocation. I barely explained any of these; the best I could do with LSI and LDA was handwaving and heuristics, and as for correlations, I rushed through them somewhere at midterm. When my students tried looking at word-correlations themselves, I realized I'd failed to convey how to use them effectively; I hadn't been clear about the pitfalls of trying to interpret correlations between frequencies of rare words. My students knew much more about the theoretical problems of quantification in general than about the *methodological* problems of putting numbers (and computers) to work effectively.

In squeezing these quite disparate elements into the syllabus—and consequently scanting each of them for time—I was trying to answer a curricular imperative to provide a useful DH overview and a course satisfying a "theory" distribution requirement. In one form or another, my dilemma ought to be familiar to almost anyone who has tried to teach a DH course: since there is typically just one such course in the curriculum, it's necessary to offer some general DH overview alongside more specific topics. But quantitative method is a bad fit for the "patterned isolation" of the literary curriculum: even more than a literary-historical period, it does not constitute a free-standing topic. And the frequent disjunctions between the theory-seminar half of my course and the practicum parts suggest that the "theory course" is not an ideal home for quantitative method either.

Better models for the pedagogical rationale of quantitative methods are found in the other intensively skills-based courses in graduate education: premodern languages and bibliography. Old English is taught in many graduate departments, and language-

20. Wickham's exposition of the grammar in the context of his software package in *ggplot2* is much more accessible than the challengingly abstract original formulation by Leland Wilkinson in *The Grammar of Graphics* (both on my syllabus, though Wilkinson was optional, and, it turned out, avoided by all my students).

learning is usually the primary focus of the first-semester course, not debates in Anglo-Saxon studies, which are introduced in a later semester. This pedagogical division does *not* mean Anglo-Saxonists are tragically untheoretical or have falsely separated the organically unified tasks of studying grammar and interpreting monster-slaying.[21] It means that their subject matter is distinctive—and cumulative—in ways that make a difference to the organization of the curriculum. Teachers of quantitative method have the same case to make for the distinctiveness of their subject matter.

Leaving both the breadth of the DH overview and the conceptual debates of theory for other courses, the quantitative analysis course (or, better, sequence) could focus on demonstrating, explaining, and above all practicing the fundamentals of exploratory and inferential data analysis on carefully curated data of known scholarly interest. There is no way to learn the required sort of logical problem-solving except by deliberate practice, but solving problems efficiently requires ways of working that students cannot be expected to know in advance. At the start of term I thought I needed only to set my students reasonable tasks and let them figure them out together. But eventually I learned that the problem sets I had written to take two or three hours were costing my students a dozen hours a week. It seems that, lacking training in solving problems in this domain, my students needed a great deal of time for trial and error.

How to teach better problem-solving technique? At a minimum, students need to see problem-solving demonstrated, then to work through a problem together with an instructor, then to practice it some more on their own. The once-weekly three-hour seminar meeting was not an ideal format for this. In a sense the class was meeting much more than that, not only in my office hours, but also when the students met without me to do the homework. A certain amount of the latter is very desirable, but really a TA would have helped a great deal: my students needed help from someone who was skilled enough to spend time doing the homework with them and reinforcing its lessons, and I needed someone to split the job of answering e-mail with. Just as learning quantitative methods is more work than fits into a single course, so is teaching them.

TOWARDS METHOD

I have argued that teaching quantitative methods is hard, but I am not suggesting that it needs to be made easy. On the contrary, DH should be wary of promises of ease: in prepackaged tools, in well-meaning introductory tutorials and workshops that necessarily stop short of what a researcher would need to draw conclusions, in rationalizations of inconclusive arguments as exploration, play, or productive failure. Having organized workshops, produced tools, and talked elsewhere about data-exploration myself, I'd like

21. Yacking and hacking, if you will. More seriously, on the meaning of the "hack vs. yack" controversy in DH, see Natalia Cecire's compelling "Introduction: Theory and the Virtues of Digital Humanities."

to think the desire for ease is understandable. These approaches can be useful in making the unfamiliar more accessible. Nonetheless, I learned from my students to be more skeptical that the easeful way can naturally lead to the production of new knowledge, because a gentle introduction simply cannot get very far in developing ways of interpreting quantitative results. Good programming skills actually exacerbate the problem when they make highly complex transformations of data—matrices of frequency correlations, probabilistic topic models, and so on—simpler than figuring out whether those transformations are appropriate to the data at hand.

Indeed, the first two problems of quantitative-methods pedagogy I have described become most evident in the context of the third: if introductory programming with arbitrary yet convenient data were all methodology required, the subject would indeed fit in a one-semester course.[22] On the other hand, the prospect of humanities departments offering multiple semesters of coursework in quantitative methods seems fanciful even to me: who would teach such courses, and how many students would really take them? Humanities students who wish a fuller quantitative training are more likely to find it in the methods courses offered by social-science or statistics departments, provided humanities departments recognize this as a possibility for credit within a degree program. Nonetheless, courses taught by humanities faculty can lay a meaningful foundation in data analysis, as long as they spend enough time introducing the problems of quantitative methodology *as* problems rather than passing them by, or, worse, imagining they have been solved because students are learning to code.

The curricular practicalities can be addressed only once the true challenges of quantitative method are more widely understood. As long as confusions between programming and quantitative data analysis reign, students and researchers alike may not even recognize when their problems are methodological rather than merely technological, and they will often be forced to stop short of making convincing arguments. Though my and my students' experience was chastening in some ways, my students' determined commitment to rigor gives me hope. Despite the total unfamiliarity of much of the course practicum, my students hung together and mastered some complex skills; furthermore, they embraced the possibility of unfamiliar ways of producing knowledge, and, in the final reports, they wrestled ingeniously with the question of how to use numbers as part of their answers to research questions. In order to go further, what we needed was, above all, more models of how to take those questions on—not just for seminar discussion as exempla of "DH" but as paths from evidence to analysis we could retrace and then continue ourselves. If research provides more such pathways, including the evidence they start from, the challenges of quantitative method could be more clearly distinguished from those of programming and other dimensions of computer-assisted hu-

22. I thank Taylor Arnold for pointing out this connection among the three problems.

manities. Then, I believe, a more focused quantitative literary-studies curriculum would become, not easy, but just hard enough.

BIBLIOGRAPHY

Arnold, Taylor, and Lauren Tilton. *Humanities Data in R*. New York: Springer, 2015.

Boltanski, Luc, and Eve Chiapello. *The New Spirit of Capitalism,* trans. Gregory Elliott. London: Verso, 2005.

Burrows, J. F. *Computation into Criticism: A Study of Jane Austen's Novels and an Experiment in Method*. Oxford: Clarendon, 1987.

Cecire, Natalia. "Introduction: Theory and the Virtues of Digital Humanities." *Journal of Digital Humanities* 1, no. 1 (Winter 2011). http://journalofdigitalhumanities.org/1-1/introduction-theory-and-the-virtues-of-digital-humanities-by-natalia-cecire/.

Crymble, Adam, Fred Gibbs, Allison Hegel, Caleb McDaniel, Ian Milligan, Miriam Posner, Evan Taparata, and Jeri Wieringa, eds. *The Programming Historian.* 2nd ed. 2016. http://programminghistorian.org/.

Darnton, Robert. "Reading, Writing, and Publishing in Eighteenth-Century France: A Case Study in the Sociology of Literature." *Daedalus* 100, no. 1 (1971): 214–56.

Graff, Gerald. *Professing Literature: An Institutional History*. 20th anniversary ed. Chicago: University of Chicago Press, 2007.

Graham, Shawn, Milligan, Ian, and Weingart, Scott. *Exploring Big Historical Data: The Historian's Macroscope*. London: Imperial College Press, 2015.

Guttag, John. *Introduction to Computation and Programming Using Python*. Rev. ed. Cambridge: MIT Press, 2013.

Jockers, Matthew L. *Macroanalysis: Digital Methods and Literary History*. Urbana: University of Illinois Press, 2013.

———. *Text Analysis with R for Students of Literature*. New York: Springer, 2014.

Karsdorp, Folgert. *Python Programming for the Humanities*. N.d. http://fbkarsdorp.github.io/python-course/.

Kirschenbaum, Matthew G. "What Is 'Digital Humanities' and Why Are They Saying Such Terrible Things about It?" Differences 25, no. 1 (2014): 46–53.

Klein, Lauren F. "The Image of Absence: Archival Silence, Data Visualization, and James Hemings." *American Literature* 85, no. 4 (December 2013): 661–88.

Lévi-Strauss, Claude. "The Structural Study of Myth." *The Journal of American Folklore* 68, no. 270 (October 1955): 428–44.

Long, Hoyt, and Richard Jean So. "Literary Pattern Recognition: Modernism between Close Reading and Machine Learning." *Critical Inquiry* 42, no. 2 (Winter 2016): 235—267.

Manovich, Lev. "What Is Visualisation?" *Visual Studies* 26, no. 1 (2011): 36–49.

McGann, Jerome J. *Radiant Textuality: Literature After the World Wide Web*. New York: Palgrave, 2001.

McKenzie, D. F. "Printers of the Mind: Some Notes on Bibliographical Theories and Printing-House Practices." *Studies in Bibliography* 22 (1969): 1–75.

McPherson, Tara. "Why Are the Digital Humanities So White? or Thinking the Histories of Race and Computation." In *Debates in the Digital Humanities*, ed. Matthew K. Gold, 139–160. Minneapolis: University of Minnesota Press, 2012.

Moretti, Franco. "Style, Inc. Reflections on Seven Thousand Titles (British Novels, 1740–1850)." *Critical Inquiry* 36, no. 1, Autumn (2009).

Ramsay, Steven. "Programming with Humanists: Reflections on Raising an Army of Hacker-Scholars in the Digital Humanities." In *Digital Humanities Pedagogy: Practices, Principles and Politics*, ed. Brett D. Hirsch, chap. 9. N.p.: Open Book, 2012. http://www.openbookpublishers.com/htmlreader/DHP/chap09.html.

———. *Reading Machines: Toward an Algorithmic Criticism*. Urbana: University of Illinois Press, 2011.

Tukey, John W. *Exploratory Data Anaysis*. Reading, Mass.: Addison-Wesley, 1977.

Wickham, Hadley. *ggplot2: Elegant Graphics for Data Analysis*. New York: Springer, 2009.

Wilkens, Matthew. "The Geographic Imagination of Civil War-Era American Fiction." *American Literary History* 25, no. 4 (2013): 803–40.

Wilkinson, Leland. *The Grammar of Graphics*. 2nd ed. New York: Springer, 2005.