Taylor & Francis
Taylor & Francis Group

Journal of Statistics and Data Science Education

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ujse21

Teaching Creative and Practical Data Science at Scale

Thomas Donoghue, Bradley Voytek & Shannon E. Ellis

To cite this article: Thomas Donoghue, Bradley Voytek & Shannon E. Ellis (2021) Teaching Creative and Practical Data Science at Scale, Journal of Statistics and Data Science Education, 29:sup1, S27-S39, DOI: 10.1080/10691898.2020.1860725

To link to this article: https://doi.org/10.1080/10691898.2020.1860725

Published online: 22 Mar 2021.

JOURNAL OF STATISTICS AND DATA SCIENCE EDUCATION
2021, VOL. 29, NO. S1, S27–S39
https://doi.org/10.1080/10691898.2020.1860725

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Teaching Creative and Practical Data Science at Scale

Thomas Donoghue[a], Bradley Voytek[a,b,c], and Shannon E. Ellis[a,b]

[a]Department of Cognitive Science, University of California, San Diego, La Jolla, CA; [b]Halıcıoğlu Data Science Institute, University of California, San Diego, La Jolla, CA; [c]Neurosciences Graduate Program, University of California, San Diego, La Jolla, CA

ABSTRACT

Nolan and Temple Lang's *Computing in the Statistics Curricula* (2010) advocated for a shift in statistical education to broadly include computing. In the time since, individuals with training in both computing and statistics have become increasingly employable in the burgeoning data science field. In response, universities have developed new courses and programs to meet the growing demand for data science education. To address this demand, we created Data Science in Practice, a large-enrollment undergraduate course. Here, we present our goals for teaching this course, including: (1) conceptualizing data science as creative problem solving, with a focus on project-based learning, (2) prioritizing practical application, teaching and using standardized tools and best practices, and (3) scaling education through coursework that enables hands-on and classroom learning in a large-enrollment course. Throughout this course we also emphasize social context and data ethics to best prepare students for the interdisciplinary and impactful nature of their work. We highlight creative problem solving and strategies for teaching automation-resilient skills, while providing students the opportunity to create a unique data science project that demonstrates their technical and creative capacities.

KEYWORDS
Computing; Course design; Data science; Programming; Project-based learning; Python

1. Introduction

In *Computing in the Statistics Curricula*, Nolan and Temple Lang made a call for a radical revamping of statistics toward accepting, adopting, and embracing computation (Nolan and Temple Lang 2010). This included treating statistics as flexible problem solving and considering the role of the statistician to include not just statistical analysis, but to more broadly include tasks such as accessing databases, designing investigations, and creating visualizations. They also issued a warning: that if statistics did not move toward incorporating these suggestions, other fields would step in and fill the void in computational science and data analysis. Their fears have at least partly come true, as their stated vision for the future seems less descriptive of a modern statistics department and perhaps closer to the emerging (though overlapping) field of data science.

We now look back on Nolan and Temple Lang's paper as predictive of many of the changes that have occurred, though often under the label of data science rather than statistics. These changes include a move toward, and proliferation of, computational work, wherein statistical analyses rely heavily on the use of programming languages and computational tools. Additionally, practitioners of data science are expected to be involved in the whole process, from formulating questions to procuring data, running analyses, and creating visualizations. This shift has led to a rapid rise of data science courses, both in universities and online. These new courses aim to implement hands-on practical training that, in many ways, mimics the vision that Nolan and Temple Lang imagined should happen within statistics.

We are instructors involved in designing and teaching new, large courses on data science that aim to teach hands-on data analysis through computational work. Building on the lessons learned from having run these courses for several thousand undergraduate students and on the work of other data science educators (Baumer 2015; Carver et al. 2016; Hicks and Irizarry 2018; National Academies of Sciences, Engineering and Medicine 2018; Wood et al. 2018; Loy, Kuiper, and Chihara 2019; Yan and Davis 2019; Çetinkaya-Rundel and Ellison 2020), in this article we discuss our approach to, and goals for, teaching data science. In these goals we find much alignment with the vision of Nolan and Temple Lang, including their positioning of statistics (or data analysis more broadly) as flexible problem solving, and in their call for pedagogy of data analytical skills to be grounded in practical work with real datasets, to which we also add a goal of scaling data education to meet demand.

1.1. *The Growth of Data Science*

Very much in line with what Nolan and Temple Lang envisioned, there is incredible demand from students interested in obtaining an education that will enable them to succeed in technical, data-related applications (Salian 2017; Tate 2017; Russell 2018; Doucette 2019). Modern work across science, technology, engineering, and math (STEM) fields, the social sciences, and throughout industry is becoming increasingly data- and computationally intense. Within industry, the demand for creative,

CONTACT Shannon E. Ellis sellis@ucsd.edu Department of Cognitive Science, University of California, San Diego, La Jolla, CA 92093.

technically skilled data scientists in the United States is growing at such a rapid pace that there is serious concern that current educational approaches cannot adapt quickly enough to fill this demand for technical data science jobs (Markow et al. 2017). As educators, it is our responsibility to provide students with a quality education that enables them to be successful in their chosen field, while also providing them the tools (both technological and intellectual) to critically engage with the real-world impact that large-scale data analyses can have. This requires providing data science education at a scale that could likely not have been appreciated a decade ago.

Data science is a fast-paced field and change is inevitable. Teaching students that data scientists are expected to be flexible and learn new skills is essential. Learning new tools and skills is, in part, a process of learning the vocabulary and knowledge needed to productively navigate available resources, search for required information, and discern a given resource's quality. Part of the instructor's role is to guide and demonstrate this behavior, helping students to navigate what is already available and learn general strategies they can employ toward learning the next tool. This includes using common tools, rather than outdated, proprietary, or niche technologies that are a disservice to students. For these reasons, we strongly encourage all new course designers to think seriously about tooling.

There is also a growing concern that technical skill alone is insufficient for long-term data science career success. This is partially due to the fact that many data science tasks are being automated (Schlegel et al. 2016; Berthold 2019). For example, it has been argued that to stay employed, data scientists should focus on developing skills that are harder to automate, like business understanding, explanation, and storytelling (Piatetsky and Gandhi 2018). Thus, while it is critical to train students in analytical skills and domain expertise, we propose also prioritizing creative, automation-resilient data science skills.

The demand for skilled data scientists is met by large numbers of students who are interested in learning and practicing data science. This recent growth in demand for data science education has come from a broader population than has been historically served by computationally-intensive fields. This interest provides data science educators with the opportunity to embrace and effectively educate individuals from groups that have historically been underrepresented in technical fields, such as computer science (Bours 2018) or engineering (Rivers 2017). The onus is on data science to ensure that it adopts equitable teaching practices and encourages a culture that welcomes and supports diverse student populations. By embracing data science education and planning courses that are inclusive and accessible (Lue 2019), we have the opportunity to serve large swaths of the population while addressing diversity issues that have plagued other fields that failed to adequately address inequities when planning their path forward (Mangalindan 2014; Wilson et al. 2014). Given the incredible demand from students and industry, we must provide this education effectively to interested student populations. As well as dedicated work to promote and foster inclusive and welcoming environments, the field requires a scalability that can accommodate large student populations, to offer data science education to all who want it.

1.2. Societal Impact

The rapid growth of data science and its applications also emphasizes its impacts. While Nolan and Temple Lang accurately predicted the need for those who analyze data to be statistically sound, computationally competent, and data literate, we propose that data science education must go further to directly address the many ways in which data and its analysis have completely reshaped society and our daily lives. There has been a massive shift in the impact that students with solid statistical and computational training will have on the world. The resulting profound and long-lasting effects data scientists will have on society necessitates that data science education incorporate impactful, real-world data into all data science courses (Neumann, Hood, and Neumann 2013; Donoho 2017), highlight and teach the importance of context surrounding the data used in analysis (Carver et al. 2016; National Academies of Sciences, Engineering and Medicine 2018; Wood et al. 2018), and focus on the impacts and ethical implications of working with data in today's data-centric landscape, explicitly discussing and teaching ethical practice and frameworks throughout (Saltz, Dewar, and Heckman 2018; Baumer et al. 2020). These topics can be introduced to students through specific examples throughout the course as well as by sharing relevant and accessible literature (Angwin et al. 2016; O'Neil 2016; Hicks 2017; Eubanks 2018; Noble 2018).

1.3. Educational Goals

Beyond focusing on flexible problem solving and practical applications, as argued in *Computing in the Statistics Curricula* (Nolan and Temple Lang 2010), the integration of societal impacts and data ethics emphasizes that modern data science reflects a novel area of practice and inquiry. Data science is a field that relies on statistics, machine learning and programming as *tools*, but which is also a distinct, emerging field in its own right, whose essence is multidisciplinary, and whose pedagogy must encompass the breadth of topics, concerns, and practicalities of working with data.

It is also our goal to be able to offer data science instruction to everyone. In a data driven world, developing data literacy and awareness of data practices is a generally useful skill. Learning about and engaging with data should not be limited to those who already have or want extensive experience with math and computation. For those who have interest and require those skills, there are dedicated courses for learning important and useful skills of applied math, machine learning, and statistics. A student's first course in data science, we argue, should motivate how and why data practice is an important topic and focus on the context, impacts, and practicalities of data practice. This also has the potential to motivate students who were not already inclined to go on to pursue more technical classes and applications.

We note that moving these ideas from laudatory goals to pedagogical reality is now also a question of scale, requiring us to scale course designs to match demand and ensure that all interested students are able to gain a beneficial education in data science. Overall, we propose that data science courses should: (1) conceptualize data science as creative problem solving, (2) prioritize practical application and hands-on training, and (3) scale educational practices to meet the demand. In the following sections, we first lay out our design for our data science course, as an example, and then step through each of these goals.

2. Data Science in Practice

Data Science in Practice is a large, project-based, undergraduate course that has been offered at UC San Diego since 2017. Course materials are managed and available through GitHub (*https://github.com/COGS108*) and are also publicly available (*https://datascienceinpractice.github.io/*). Specifics of the materials are further described in the publicly facing repository (*https://github.com/DataScienceInPractice/Site*). In this section, we briefly introduce the student population, course design, and course content, which we present as an example for the pedagogical goals discussed throughout the article.

2.1. Student Population

This course has high student demand, being offered nearly every term, with at least 400 students in every iteration and more than 3300 students having completed the course in its first 3 years. As an upper-division course, students in Data Science in Practice tend to have third (∼36%) or fourth (∼58%) year standing (of a four-year degree) upon enrollment. Students come from a wide variety of majors and educational backgrounds, with the only prerequisite being at least one class of programming experience. In a typical offering, there are students from approximately 15 different majors; however, the three most common majors of students in this course are Cognitive Science (∼30%), Computer Science & Engineering (∼29%), and Data Science (∼21%). Typically, ∼40% of students are female. The course is run by a relatively small teaching staff of approximately 110 students per teaching staff member (the professor and teaching assistants).

2.2. Course Design

The core goal of the course is to provide students with hands-on training in working with data. The structure of the course is designed to introduce students to the skills and practicalities of doing modern data science and to provide students with opportunities to learn how to develop questions and then answer them with data. The organization of the course includes four structural elements: lectures, lab sections, assignments, and a group project. Topics and skills are introduced through lectures and tutorials, implemented and practiced through individual coursework, and then brought together in the project. Collectively this design aims to ensure that students who complete this course will be able to: (1) work with data to creatively solve problems, (2) carry out this work using tools and procedures common to data science practitioners, and (3) obtain this knowledge regardless of class size.

2.2.1. Lectures

Lectures are used as the initial point of contact with the course material. Lectures are designed to introduce core data science concepts and examples, and to provide students with the fundamental background knowledge and context required of data scientists. To improve accessibility, lectures are recorded and made available to students online.

2.2.2. Lab Sections

Weekly lab sections build upon lecture material by providing students with the opportunity to work through technical material. Teaching assistants lead guided tutorials through hands-on materials relating to the topics covered in lecture that week. Lab is used to build upon concepts introduced in lecture, to solidify understanding, and to explore and apply the skills and concepts discussed in class using minimal datasets and simple case studies. Questions and examples often closely mimic those to come in assignments, such that students have simplified examples of how a particular analysis is applied.

2.2.3. Assignments

Individual assignments are completed using programmatically-graded computational notebooks, in which students use code to answer prompts that guide them to gather, analyze, and interpret data. Assignments typically use multiple, large, and often messy datasets. These assignments are designed as formative assessments, in which the guided prompts seek to teach the students how to apply the concepts of the course in real code with real data, building on topics introduced in lecture and lab section, and pushing them to generalize new skills to new, larger, and more complex datasets. These assignments are used to increase students' technical fluency and to deepen their understanding of important concepts in data science within the context of addressing the practicalities and complexities of real datasets across multiple topics.

2.2.4. Group Projects

The course culminates with a final project, completed throughout the term in small groups of 4–6 individuals. Students must craft a well-defined research question and then identify and analyze real datasets to answer that question. Through the projects, there is an emphasis on crafting good research questions, finding, incorporating, and using large heterogeneous, but most importantly, appropriate datasets, and employing appropriate and interpretable analyses, rather than on technical complexity. The final product of each project is a detailed report, organized as a computational notebook, and in the style of a standard scientific report, as detailed in Table 1. Students are provided with detailed instructions and a checklist to help guide them through project completion (*https://github.com/COGS108/Projects*). The project submission includes all the code used for all components of the project, as well as written text and data visualizations. Students also submit a survey with feedback about the project and each individual's contributions.

We also employ strategies for promoting organization, time-management, and implementation of the project consistently throughout the term. We discuss the project regularly, demonstrate example projects, and discuss question formation, dataset selection, and available resources. Project proposals are submitted early in the term (usually around week 3), and must establish a timeline and expectations for the project, which serve to aid in

Table 1. Required sections for submitted project reports.

Project report
1) Research question: What is the primary analysis question?
2) Hypothesis: What is the prediction?
3) Dataset(s): What data is being used to answer the question(s)? Describe the dataset(s).
4) Background: Why is this question of interest, what background information led to the hypothesis, and why is this important?
5) Methods: What methods have been used to analyze the data?
6) Ethics and privacy: Acknowledge and address any potential ethics and privacy issues related to the project.
7) Discussion: Discuss the potential civic and social impact of the project. Discuss any problems that were encountered, and enumerate the limitations of your project.

accountability, organization, and successful project completion. Each group receives detailed feedback on their proposed project, and we prioritize staff time for feedback and consistent support for project work.

2.3. Course Content

A full listing of topics and schedules for the course are on the course syllabus, available in the course repository on GitHub (*https://github.com/COGS108/Overview*). Overall, Data Science in Practice focuses on the practicalities of doing an end-to-end data science project, including: formulating questions of interest into data questions, considering the social and ethical implications of investigations and deciding what should be pursued, finding and accessing publicly available datasets (including using application programming interfaces (APIs) and interacting with SQL databases), data wrangling (including understanding and loading various file types, such as JSON, CSV, and XML, etc.), data preprocessing and cleaning, analyzing data to answer the questions at hand, and finally interpreting, contextualizing, and communicating the results. How these topics are covered in assignments is described in Table 2.

The hands-on component of Data Science in Practice uses the Python programming language and its data science tools, including `pandas` (McKinney 2010), `numpy` (Harris et al. 2020), `matplotlib` (Hunter 2007), `seaborn` (Michael Waskom et al. 2020), `requests` (Chandra and Varanasi 2015), `BeautifulSoup` (Richardson 2007), `statsmodels` (Seabold and Perktold 2010), `scipy` (Virtanen et al. 2020), `scikit-learn` (Pedregosa et al. 2011), and `nltk` (Loper and Bird 2002), managed with the Anaconda distribution (Anaconda 2016). Throughout the course, we use computational notebooks, specifically Jupyter notebooks (Kluyver et al. 2016). Students are also taught about version control and are required to use and become familiar with git and GitHub.

To provide an example of how course content is taught in this class, we'll use the example of data wrangling. After introducing data wrangling and related tooling in lecture, lab sections use guided examples and provide practice with working with and combining heterogeneous datasets. Students are then given an assignment with four heterogeneous, simulated datasets in two different file formats (JSON, CSV): (1) user data from a dating website, (2) employee information, (3) company user data, and (4) population data. The assignment asks students to wrangle these datasets by loading, integrating, and analyzing them together. We also integrate discussion of data context and ethics by discussing and guiding them through analyses in which students work through how to identify users of the dating site from the information in the other datasets (note that these data are all simulated). After identifying personal information, students are asked to anonymize the information in their dataset according to the Safe Harbor Method (Office for Civil Rights 2012). Throughout this assignment, students work with data wrangling, heterogeneous datasets, data visualization and interpretation, and data privacy, all while working through a narrative and the implications of having access to people's data, and what kind of safeguards need to be employed (Sweeney 2015; Zang et al. 2015).

Coursework is designed to help students develop skills for successful completion of group projects. For their final projects, students are encouraged to explore topics and issues that are of particular importance or interest to their group or their community. In terms of content, this leads to a large variation of topics and approaches across the course (see the example project in Figure 1). The project continues an emphasis on social context and data ethics (Table 3) and pushes students to work collaboratively. A concrete outcome of the projects is that students finish the course with a deliverable—a digital record of their work that demonstrates their accomplishments and skills in pursuing novel questions using practical and standard data science tools and approaches.

3. Data Science as Creative Problem Solving

We organize our teaching around the primary goal of educating students in data science as conceptualized as creative problem solving. Trainees will be entering a field that is both dynamic and impactful. In addition to the statistical, technical, and scientific skills critical for success in data science, students should be trained to be resilient to the increasing automation of aspects

Table 2. Assignments in Data Science in Practice, and the main concepts, skills, and tools they cover.

	Concept(s)	Skills	Tools
A1	Version Control & Python	version control, python standard library	Jupyter Notebooks, git/GitHub
A2	Data Wrangling	data wrangling, tabular data	`pandas, numpy`
A3	Data Exploration	data wrangling, EDA, dataviz	`pandas, matplotlib`
A4	Data Privacy	heterogeneous data, web scraping, integrating datasets, reidentification	`pandas, requests, BeautifulSoup`
A5	Data Analysis	data cleaning, data wrangling, EDA, regression	`pandas, statsmodels, scipy, matplotlib`
A6	Natural Language Processing	text data, data wrangling, TF-IDF, NLP	`pandas, nltk, scikit-learn`

NOTE: The main concepts, skills, and tools covered across the assignments for a single iteration of Data Science in Practice. EDA, Exploratory Data Analysis; TF-IDF, Term Frequency-Inverse Document Frequency; NLP, Natural Language Processing; nltk, Natural Language Toolkit.

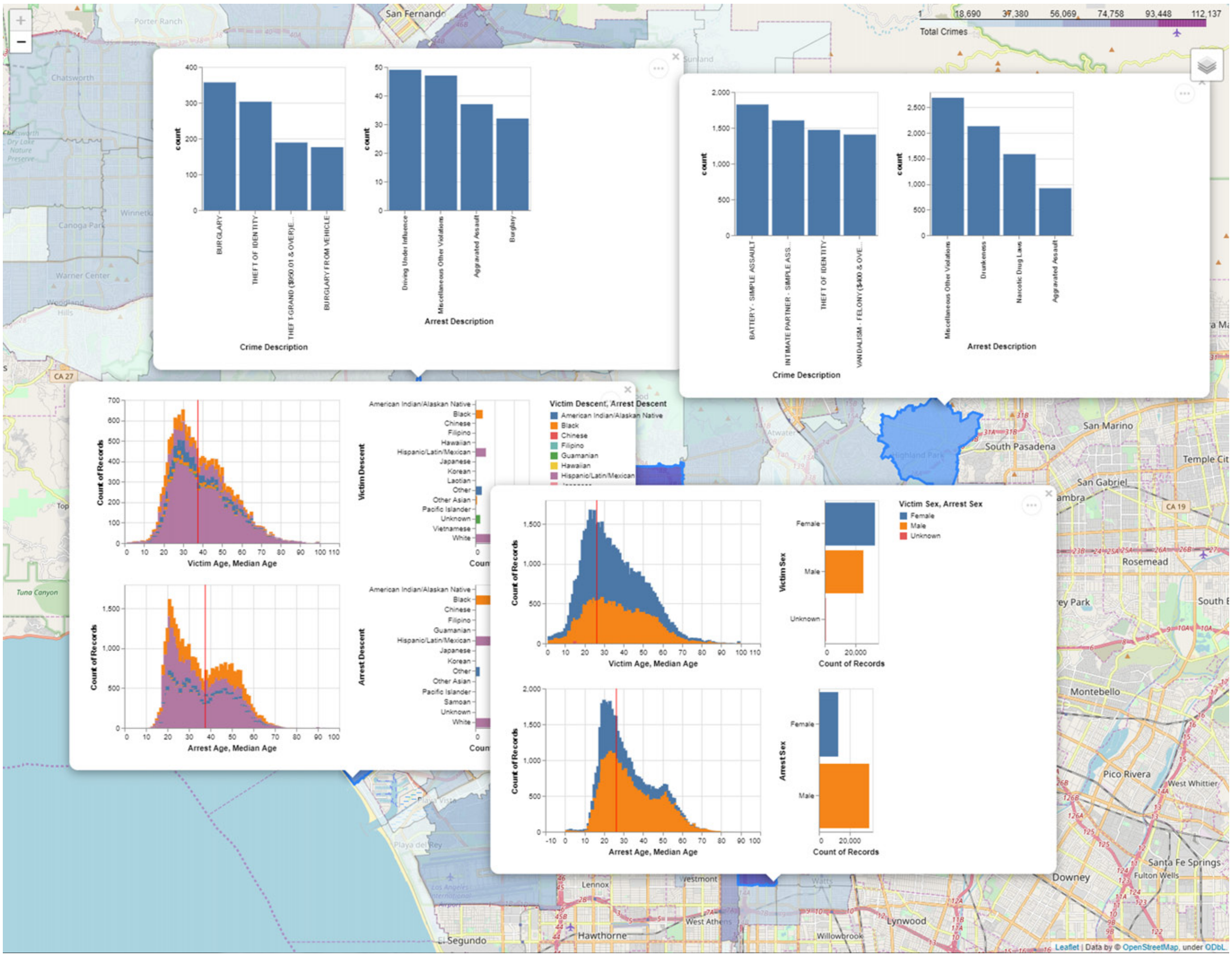

Figure 1. Screenshots from an example Jupyter notebook from a mentored group data science project from the Data Science in Practice class (Chu et al. 2019). This group analyzed publicly available crime data from the City of Los Angeles. Data were analyzed at the zip code level to examine how population, median home price, and streetlight prevalence related to crime and arrest rates.

Table 3. Ethics and privacy considerations in data science projects.

Ethics and privacy
– Do you have permission to use this data/use it for this purpose?
– Are there privacy concerns regarding your datasets that you need to deal with, and/or terms of use that you need to comply with?
– Are there potential biases in your dataset(s), in terms of who it composes, how it was collected, and when it was collected that may be problematic in terms of it allowing for equitable analysis? (For example, does your data exclude particular populations, or is it likely to reflect particular human biases in a way that could be a problem?)
– Are there any other issues related to your topic area, data, and/or analyses that are potentially problematic in terms of data privacy and equitable impact?
– How did you handle issues you identified?

of data analysis and to be aware of and responsive to the societal impacts of data-driven work.

3.1. Fostering Data Literacy

Effective data science courses must aim to educate students to answer the question "How can we use data to solve a given problem?" Being able to productively treat data as a primary candidate for answering a question or solving a problem requires data literacy, by which we mean a learned ability of being able to think with and about data. This data literacy develops as data practitioners spend time engaged with real-world datasets and their typical properties and limitations. This familiarity then allows for data practitioners to develop a catalogue of strategies and informed decision processes, or sometimes "best guesses," for how to navigate through the otherwise intractably large space of choosing which data to use and which analyses apply.

Developing data literacy helps students *learn* intuition for thinking about data properties, such as being able to anticipate likely sources of noise and missing data, predicting likely distributional properties of data, and given these properties, navigating which tools and analyses are likely to be most appropriate. Data literacy, we believe, arises not only from technical skills and knowledge, but is driven by actual experience with the mechanics of working with real-world data. That belief motivates why we prioritize teaching students the joys and frustrations of the *practice* of data science. In this course, we emphasize and develop data literacy through examples, tutorials, and assignments, in which we ask students to consider, explore, check, and visualize each dataset they work with. For example, for every plot generated during an assignment, students have to explain their interpretation of that plot or answer a multiple choice question to demonstrate understanding of the visualization and how it relates to the dataset and question at hand. With consistent practice and experience with a myriad of different visualizations and types of data, students are then well-positioned to generate and interpret plots on their own in their final projects.

3.2. Considering the Context of Data

A key component of data literacy is thinking critically *about* data (Shields 2005; Carver et al. 2016; Wood et al. 2018), including what technical and societal processes generate those data. We must engage with our students and train them to think about data—and data-driven approaches—from multiple perspectives, including discussing both intended and unintended consequences of data-centric projects. This means, for example, considering the people and processes involved in data generation and the scientific, social, and/or business implications of using these data, and being aware of how to productively use data to meet the goals of a research group or organization, while acknowledging the stakeholders and affected parties from pursuing a data-driven project. That is, we consider it a necessary and important aspect of data literacy to be aware of and to consider not just the literal aspects of dealing with data, but to be able to consider data in context—including where the data comes from, what it means, and who it affects.

With these goals in mind, we explicitly and consistently discuss data science ethics (Mason and Patil 2015; Saltz, Dewar, and Heckman 2018; Baumer et al. 2020), data privacy and anonymity (Kramer, Guillory, and Hancock 2014; Bird et al. 2019; Raji et al. 2020), and algorithmic fairness and bias (Buolamwini 2018; Buolamwini and Gebru 2018; Keyes, Hutson, and Durbin 2019). During the first week of class students are introduced to multiple examples of ways in which data science can cause harm and they are provided examples of each, including, but not limited to, asking a poorly-posed question (Wu and Zhang 2017) and failing to obtain informed consent (Kramer, Guillory, and Hancock 2014). Ethics is revisited in lectures and assignments throughout the course, and example datasets are chosen both to demonstrate inclusive and diverse datasets and/or to explicitly discuss limitations of the data and considerations, such as who might be missing. Ultimately, by the time we teach machine learning (toward the end of the term), students are able to think deeply and critically about the possible concerns that come along with data being used in machine learning and the consequences of these analyses when discussing how predictive algorithms affect our lives (Angwin et al. 2016; Keyes, Hutson, and Durbin 2019).

After consistent discussion throughout lecture and assignments, student understanding of the context of data is embodied in students' final projects. Project proposals and final reports require a section on the ethical implications of their projects (Table 3). For each dataset used, students must consider how the data were obtained, from whom they were obtained, when they were collected, how they should be analyzed, and what the implications of the analyses are. Students' initial proposals often focus more specifically on topics of privacy, such as personally-identifiable information or informed consent. Though important, this is an incomplete consideration of the context of the data they're using, the possible consequences of their work, and potential unintended consequences of their project. Students are provided feedback to guide their thinking and edit their project reports. Through this feedback, class discussions, and course materials, students develop the ability to discuss the context of data and the ethical implications of their work much more deeply upon submission of their final report.

3.3. Integration of Heterogeneous Datasets

A necessary aspect of modern data science is learning how to combine multiple, heterogeneous datasets (Wu et al. 2014; Voytek 2016). Solving real problems with data means leveraging data recorded across modalities that may consist, for example, of free-form text, categorical data, continuous variables, and temporal and/or spatial data. To be useful for a given project, all data need to be integrated. Determining which data one needs and how to effectively combine it is itself a creative process. For example, a data scientist might be tasked with combining (1) *self-reported demographics* and (2) *social networking data* with (3) *geolocated* (4) *freeform text* generated at different (5) *timepoints*. These five types of data are very different: categorical, weighted graphs, geospatial, natural language, and temporal, respectively. What is the appropriate way to aggregate and synthesize these data? What are the benefits and caveats for, say, aggregating geospatially versus temporally? Being able to conceptualize how to carry out this integration is necessary before any technical skills will be useful. Thus, students must be taught first principles for integrating heterogeneous data (Wickham 2014) and be given the opportunity to practice it, which we embody in our course by demonstrating multiple data modalities and tasking students with exploring how to combine them (see Section 2.3).

3.4. Data Communication, Visualization, and Storytelling

Turning data into action requires effective communication of the outcomes of complex, data-driven projects. Often the process and results must be communicated to nontechnical decision-makers. Students must therefore learn to present their results effectively to both technical and nontechnical audiences. When communicating results to any audience, data scientists should leverage training from both graphic design principles and storytelling elements. A data science course must teach students not only the core principles for effective information visualization (Tufte 1986; Wilkinson 1999; Cairo 2012; Yau 2013; Knaflic 2015; Nolan and Perrett 2016) but also the three critical storytelling principles: show, explain, and narrate (Krzywinski and Cairo 2013). Creativity in presentation and effective attention to design details further demonstrate the creative underpinnings of the field of data science. This thinking is also covered in explicit data communication lectures and manifested in the final report, wherein students must consider not only the analysis they've done and its implications but also the effectiveness of their written communication, the clarity in their story, and the careful design of their visualizations (see Sections 2.2 and 2.3).

3.5. Data Science as Problem Solving

Collectively, we conceptualize data science as an endeavor of creative problem solving. While data literacy includes a core set of technical competencies, this is a bedrock from which practitioners must be creative in deciding which questions to pursue, and how. Data scientists must learn to think about their data and its relation to the greater data ecosystem, and evaluate not simply how to solve problems with data, but also to evaluate the impacts of doing so. As data scientists craft questions of interest, they learn to ask questions such as, "What other publicly available datasets can I leverage to address my questions of interest?" and learn how to seek out and incorporate those datasets. Such datasets need to be integrated and be appropriately analyzed, interpreted, visualized, and communicated to relevant parties. Altogether, these are necessary components for creating practical, productive and context-aware data science projects.

This kind of problem solving—of choosing questions, designing strategies to answer them, contextualizing the results, and using visualization and narrative to come to data driven conclusions—also uses a skill set that is automation-resilient. While the specific tasks, implementations, tools, and analyses students develop through the coursework are liable to change or become increasingly automated, creative problem solving is a more resilient skill. While technical and analytical skills are of course important and form a key part of the course content, ultimately these skills must be in service to the overall ultimate goals of data science—of answering questions and solving problems with data, and all that this entails. In our coursework, group projects embody the problem solving aspect of data science, requiring creativity and tenacity to go from an open-ended project description to designing and then answering a data-driven question (see Sections 2.2 and 2.3).

4. Prioritizing Practical Application

We take the position that teaching data science courses requires instruction that not only embraces data science as a creative process but also exposes students to the day-to-day tooling and skills a working data scientist needs. Nolan and Temple Lang summarized this point in their statement that "a lack of computational reasoning skills makes it difficult for statisticians to work in a team where others are computationally capable" (Nolan and Temple Lang 2010). Taking this one step further, in today's world, computational skills are all but *required* for success in data-centric and technical fields. Accordingly, courses must use standard tools and demonstrate best practices in the field. Students who take a practical data science course should leave the course with (1) skills in a tool that is used by practitioners, (2) the ability to learn a new tool, and (3) demonstrable relevant work, such as a project, that showcases their capabilities to potential employers.

4.1. Focusing on Data

Data science courses should be designed around the focal topic of data, and not be replacements for statistics or programming courses. In statistics courses, time constraints often limit the number of messy datasets students get to work with, and often a course uses only idealized datasets. Similarly, in introductory programming courses, learning syntax and programming principles fills the curriculum, limiting the ability to focus on working with data. Before enrollment in Data Science in Practice, students have taken an introductory-level programming class (usually Python), and the vast majority have taken at least one introductory statistics course, making them well-positioned to excel in a data science course where examples and analyses leverage messy, real-world data. Pedagogical focus can then be on understanding both the limitations and opportunities of data-driven work and on technical expertise that is specific to working with real data.

In our course, data practice is embodied in consistent use of multiple, heterogeneous datasets. By having a solid computational and statistical framework, students have sufficient time to both consider and discuss the context behind the data and to wrangle the data from the raw format in which the data are provided into the format needed for analysis—requirements of every data scientist.

4.2. Prioritizing Relevant Tooling

Data scientists complete their work using a host of important tools. We encourage careful consideration around the tooling chosen for a given course prior to its first offering. The vast majority of data science job postings require experience in Python, R, SQL, or some combination thereof (Ruiz Junco 2017). In a similar vein, data scientists frequently carry out and communicate computations in computational notebook settings, such as Jupyter Notebooks (Kluyver et al. 2016) or R Markdown documents from within RStudio (Xie, Allaire, and Grolemund 2018; Allaire et al. 2019). This is also true within academia, as research moves toward general, open-source tools (Voytek 2017; Wilson et al. 2017; Wessel, Gorgolewski, and Bellec 2019). We encourage that course design should prioritize the use of these tools as students with experience in common languages and tools are immediately more employable and more competent at a variety of technical tasks.

We emphasize to students that they are not only learning current tools, but also learning strategies to navigate resources and future tools. We emphasize that learning this kind of navigation is a common experience for experts as well—that part of expertise is not necessarily just knowing the answer, but rather, knowing how to find it. Data Science in Practice is focused on data-centric programming—programming in which the primary function is the management and manipulation of data—using the Python programming language and ecosystem. Course materials and projects are also managed on GitHub, requiring students to gain fluency in version control. Students also finish the course with their project as an example of their work that demonstrates their knowledge of relevant tools.

4.3. Implementing Best Practices

To carry out the breadth of a data science project efficiently and effectively, data scientists need to employ best practices. Best practices for data science include project management, version control, reproducibility, and data-centric programming, all of which should be required in data science courses. Following best practices requires data scientists' work to be reproducible, which demands that given the data and code, a different analyst would be able to arrive at the exact same conclusions as the original analyst (Patil, Peng, and Leek 2016). This requires that data scientists move beyond simply sharing their code to ensuring the code is well-organized, documented, and tested (Taschuk and Wilson 2017; Wilson et al. 2014, 2017). To facilitate this, data scientists must demonstrate their ability to manage, organize, and carry out a project. This includes, but is not limited to: organizing the project's file structure, storing the raw data in a consistent manner, writing code that is well-documented and understandable by other analysts, and testing and reviewing the code and its outputs.

Project management in data science differs from how a computer scientist manages a software development project, just as code written to analyze a dataset differs from code written during software development. Given these differences and nuances specific to data science, explicit instruction in these topics is essential. Students should be provided examples of data science projects and given opportunities to practice managing projects. Within projects, students must have opportunities to write good and well-documented code that accomplishes their analysis goals (Wilson et al. 2014, 2017). Throughout this entire process, version control should be used to facilitate collaboration with other students, code sharing, and portfolio generation (Blischak, Davenport, and Wilson 2016; Çetinkaya-Rundel and Ellison 2020).

In our course, best practices are demonstrated throughout, exposing students to well-commented and clean code that demonstrates best practices in data analysis. Students are given the opportunity to write good code and carry out meaningful analyses throughout their assignments and projects. The final project encourages students to demonstrate project management skills, using version control and collaborating with their group mates, resulting in well-written and reproducible code to answer their question of interest.

4.4. Debugging (or "Teaching How to Get Unstuck")

Navigating real datasets, with a large set of possible tools and analysis directions, is a massive space to explore. Data science courses must teach students that a key to being a productive programmer and data analyst is having strategies for debugging issues when they arise. Key skills for the budding data scientist include how to explore and debug both code and data issues, and how to decide on a path forward when what to do next is unclear. In terms of data analysis specifically, this requires a set of problem solving skills to investigate data, algorithms, and method assumptions to productively explore possibilities if and when things are not working as expected. These skills—of debugging and data problem solving—are core skills for the data scientist, though they are not always explicitly taught as such.

We seek to explicitly instruct students on the data-centric debugging strategies employed when analyzing data by running sessions on debugging and how to proceed if one's code is not working. In these sessions, we emphasize strategies for data exploration that can be used to audit data properties and explore problems. Importantly, these should be emphasized as strategies that should be employed whether or not there is an explicit error present. This helps to promote robust data analysis that is sensitive not only to egregious and obvious errors, but that maximizes the chance of discovering silent but impactful errors. This idea can also be embodied by showing students how to answer a question, rather than simply offering an answer. For example, when students ask how to accomplish a particular task, we will often talk through our thought process while demonstrating our search procedure for finding appropriate tools or documentation, and recommend they take a similar approach.

4.5. Developing Collaborative Practices

Teaching practical tools and best practices for data science is designed to give each student fluency with the concepts and skills they will need for the future. However, it's also important to note that the practice of data science is rarely, if ever, an individual endeavor. Whether in academia or industry, data-driven projects typically require coordinated work across multiple individuals within technical teams, as well as interaction and collaboration with interested parties and stakeholders. This requires collaboration across teams in which people come from different backgrounds and have varying skill sets and specialties. Interdisciplinary teams are effective (Mangalindan 2014; Nelson 2014), but productively working within them is also a skill that can and should be taught and practiced in data science courses.

Collaborative practice can be modeled with team-oriented project work (see Sections 2.2 and 2.3), which has numerous, well-documented pedagogical benefits (Williams, Beard, and Rymer 1991). In contrast to assignments (which are necessarily guided), mentored group work provides students flexibility in thinking, requires creativity, and allows for self-directed work. Projects allow students to work on a topic that they are particularly passionate about and to engage in problem solving activities more similarly to how they may do so on the job (Williams, Beard, and Rymer 1991). To promote interdisciplinary interaction, students can be encouraged to form groups with classmates from varied backgrounds, and even to organize group work according to individual specialties. In our courses, students come from a variety of majors, which helps to promote interdisciplinary work. For example, in a recent iteration with self-selecting groups, 55% of groups had students from at least three different majors, and only 8% consisted of students from a single major. These team-oriented projects provide students with the opportunity to share and develop their expertise with each other and practice collaborative work.

5. Teaching Data Science at Scale

Scientific research is becoming increasingly dependent on increasingly complex computations applied to increasingly larger datasets (Lohr 2012; Peters 2012), while at the same time, the growth of data-driven analytics in industry is also creating a high demand for trained candidates (Markow et al. 2017). The generality and broad interest in data science topics is creating high demand, which, in turn, necessitates designing and teaching large data science courses. These large enrollments make it difficult to teach courses focused on creative problem solving within the context of practical and applied problems. Course designs that accomplish what we have described thus far would most typically be small, project-based courses. When faced with increasing enrollment, most common strategies for scaling classes often veer toward rote learning and abstract, decontextualized practice. However, we consider such an approach antithetical to effectively teaching data science.

The difficulty is how to scale the types of learning and experience that we wish to offer, in the context of necessarily large classes and limited resources for instructors and teaching assistants, as is typical at public institutions such as our own. Luckily, the same proliferation of computation and tools that has prompted the rapid rise of data science has also enabled advances in education technology and open-source resources that offer possible solutions to this scaling problem. In this section we will discuss tools and strategies we have employed to offer productive and efficient hands-on instruction to large classes with minimal teaching staff.

5.1. Computing Environments

Using cloud environments for students' computation can assist with scaling computation-based courses (Çetinkaya-Rundel and Ellison 2020). Compared to other options, this solves or avoids problems of getting software set up on individual or university computers, such as: ensuring the same software environment across all users, releasing and collecting course materials and assignments, and ensuring that all students have access to equivalent computational resources. Recent iterations of our course have used a JupyterHub instance (*JupyterHub* 2017) administered and managed by the university IT department. Institution-level resources, when available, likely offer the most scalable option for courses at the level of hundreds of students, though individual JupyterHub instances or similar can be set up for smaller classes.

We do note that having students install and run software locally is possible. In earlier iterations of the course, students each installed software environments and did work on their own computers. This is manageable if students have computers available and clear instructions are provided. In our case, we required the use of the Anaconda distribution of Python, with no extra dependencies, and provided detailed instructions for and help with installation. Using local installs has the added benefit that it ensures students can run software on their own system, not being limited to external systems, such that they may be more able to continue independent work after access to the cloud resource is lost.

5.2. Course Organization

There are several general practices in higher education that assist in scaling courses. In lectures, we employ existing tools and strategies, such as using clickers to poll students with check-in and peer-instruction questions (Porter et al. 2016). Lectures often feature live demos (Rubin 2013) and tools such as Google Forms can be used to collect responses, such that students can experience and explore issues related to data collection (i.e., free-form text entry vs. multiple choice selection). To further foster active learning, polls, demonstrations, discussions, and other strategies that increase interaction can help to increase engagement in course material, even with large class sizes (Freeman et al. 2014; Prince 2004).

With limited time available from instructors, optimizing this time to prioritize productive interactions with students is key. With ample materials, recorded lectures, easy access to information, and numerous examples and links to external resources all clearly organized and available through the materials hosted on the GitHub organization, the goal is that the most common questions and use cases can be answered through interactions with the course materials. Lab sections, as well as office hours, which are run as open-work time in computer labs, offer time for more detailed questions and direct feedback from the course staff. Messaging boards can be used to minimize E-mail load and allow students to see and answer each other's questions. We also make sure to promote data science as a collaborative endeavor and encourage students to assist each other with learning concepts, working on code implementations, and with project work.

A benefit of teaching data science as it emerges as a massively popular topic is that many other educators, institutions, and organizations are currently involved in creating and promoting materials, many of which are freely available. This provides many available resources for instructors to draw from, and to direct students toward, which can often offload the creation of new materials from course staff. Throughout the course, we aim to direct students on how to become proficient navigators of the data science environment. This strategy helps to teach students the "map" of what is available and helps to empower them to be more self-sufficient learners. For example, packages such as `pandas` (McKinney 2010) have entire ecosystems of materials, much beyond what we can cover in a small number of lectures, and so we consistently remind students to search for additional materials and descriptions of additional functionality that may be useful as they encounter new questions and problems, for example in their projects. This promotes better strategies for dealing with problems, reducing the need for instructor intervention, and developing strategies so students can continue to learn after finishing the course.

5.3. Scaling Course-Work

In courses with hundreds of students, having multiple individual assignments quickly generates thousands of completed assignments per term, at an intractable scale for manual grading. There are existing tools that serve as automated grading systems, for example the `nbgrader` tool for Jupyter notebooks (Project Jupyter et al. 2019). This system allows instructors to write computation-based, data-centric, guided assignments, with accompanying prewritten test code that checks each student's answers against specified criteria and awards points accordingly.

Designing and building stable and productive assignments is a challenging and time-consuming task, with some limitations. Each question has to be preformulated in a way that specific test cases can be written out ahead of time. This format does not lend itself to asking students to design and implement their own analyses, but is instead focused on prompting students to use and work through standard approaches and provide practice with necessary skills. Care does need to be taken to fully describe questions, and code tests, as autograded questions are only as good as the prewritten test code that grades them. The benefit, once an investment has been made to develop well-formed assignments, is that they can scale to develop procedural skills and offer practice for large numbers of students, and automated feedback can be used to address common misinterpretations.

5.4. Scaling Project Work

Perhaps the most challenging aspect of scaling course designs to teach practical, hands-on data science lies in the implementation of group projects. In our courses, there is no single element that makes this work, but we have found that many coordinated strategies throughout the course can make this possible. In part, this stems from organization throughout the course to use demonstrations, examples, and hands-on practice so that students become more independent and develop strategies for navigating problems as they arise. Using computational tools and automation assists in the practical aspects of managing large numbers of projects. After receiving their proposals, automated scripts are used to generate project repositories for each group on GitHub, with students subsequently being required to coordinate their project using git. As well as serving as practice for group work using a version control system, this approach means that course staff have access to their materials, making it easier for students to ask questions and get feedback remotely.

By making all assignments autograded, we can heavily prioritize dedicated instructor time to project work. Because of the personalized nature of the projects, they are the only manually-graded aspect of the course. Course staff allocate significant time to project-related feedback and grading, with a priority given to providing explicit feedback on project proposals. Organization and best practices for pedagogy also assist with scaling courses efficiently and equitably. Grading rubrics are used to assist in ensuring consistency across staff members. Projects are explicitly collaborative, and we work throughout the term to foster a collaborative environment by establishing ground rules for an inclusive classroom. We encourage students to help each other with project work, run open group office hours, and frequently dedicate lecture time to meaningful interaction between the instructors and students, rather than constant lecture. Though still challenging, by setting priorities and allocating time accordingly, project work can be scaled to large courses.

6. Discussion and Conclusion

Nolan and Temple Lang's *Computing in the Statistics Curricula* (Nolan and Temple Lang 2010) insightfully called for a revamping of statistics education to require and embrace computing, strengthening its ties to computational reasoning and literacy, and prioritizing the use of data. In this work, we have continued this discussion of teaching practical problem solving with data. In the context of modern data science, this education must be accessible to all and seek to contextualize and address the impacts of data, teaching how to think both critically and creatively, while prioritizing generalizable skills in the face of an ever-changing field. To accomplish this we set out three pedagogical goals. First, to teach data science as creative problem solving, providing students with skills that require critical thinking and that are automation-resistant. Second, to design content and materials around practical application, requiring students to use relevant tooling and best practices in the field. Finally, to scale courses to ensure that this education can meet demand and be accomplished in large-enrollment courses through the use of individual assignments, group projects, and educational tooling.

As an example, we introduced our course, Data Science in Practice, in which we are educating hundreds of undergraduates each year in data science with limited instructional staff. Despite the large numbers, we aim to ensure that every student is able to learn, practice, and demonstrate practical skills such as data literacy, data intuition, creativity, and problem solving, as well as contextual understanding of their work. Throughout this course, we ask students to become familiar and comfortable with developing questions and then answering them with data, while acknowledging the limitations and impacts of their work.

In working toward these goals, we note there are a number of limitations left unaddressed thus far. From a course design perspective, group work necessarily leaves some students doing and learning more than others. We do our best to mitigate this, soliciting feedback at the end of the quarter from all students about their group work experience, adjusting grades as needed. Similarly, to focus on hands-on work, we do not have exams in this course. This allows some students to fall behind on course material, limiting their knowledge for the final project. We've recently included weekly quizzes that contribute to a small portion of students' final grades to mitigate this. Each student also receives submission-specific feedback on their assignments and each group receives group-specific feedback on their projects. Finally, students have the opportunity to gain more individualized attention in labs and instructional staff office hours to help ensure that no student falls behind.

The growth of data science is going to continue, and, as educators, we must work to meet the growing demand and develop education that scales. In doing so, we envision data science moving in the direction of creative problem solving, where students need to be trained not only to be statistically sound, computationally competent, and data literate but also to develop the data skills, intuition, and creativity that will help them to be continuously successful. As educators, we must acknowledge the power and opportunity, but also the jeopardy, that stems from the multi-disciplinary endeavor of working with data. We must work accordingly to provide our students with effective pedagogy to guide them to become the careful and capable practitioners of data science that we need.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Funding

Voytek is supported by the National Science Foundation under grant DGE NRT 1735234 and a Halıcıoğlu Data Science Institute Fellowship.

ORCID

Thomas Donoghue http://orcid.org/0000-0001-5911-0472
Bradley Voytek http://orcid.org/0000-0003-1640-2525
Shannon Ellis http://orcid.org/0000-0002-9231-0481

References

Allaire, J. J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., and Iannone, R. (2019), "rmarkdown: Dynamic Documents for R," available at *https://rmarkdown.rstudio.com/docs/index.html*.

Anaconda (2016), "Anaconda Software Distribution," available at *https://anaconda.com*.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016), "Machine Bias," *ProPublica*, available at *https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing*.

Baumer, B. (2015), "A Data Science Course for Undergraduates: Thinking With Data," *The American Statistician*, 69, 334–342, DOI: 10.1080/00031305.2015.1081105.

Baumer, B. S., Garcia, R. L., Kim, A. Y., Kinnaird, K. M., and Ott, M. Q. (2020), "Integrating Data Science Ethics Into an Undergraduate Major," arXiv no. 2001.07649.

Berthold, M. R. (2019), "What Does It Take to be a Successful Data Scientist?," *Harvard Data Science Review*, 1, DOI: 10.1162/99608f92.e0eaabfc.

Bird, S., Kenthapadi, K., Kiciman, E., and Mitchell, M. (2019), "Fairness-Aware Machine Learning: Practical Challenges and Lessons Learned," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM'19*, Association for Computing Machinery, New York, NY, USA, pp. 834–835, DOI: 10.1145/3289600.3291383.

Blischak, J. D., Davenport, E. R., and Wilson, G. (2016), "A Quick Introduction to Version Control With Git and GitHub," *PLoS Computational Biology*, 12, e1004668, DOI: 10.1371/journal.pcbi.1004668.

Bours, B. (2018), "Women and Minorities in Tech, by the Numbers," *Wired*, available at *https://www.wired.com/story/computer-science-graduates-diversity/*.

Buolamwini, J. (2018), "Opinion | When the Robot Doesn't See Dark Skin," *The New York Times*, available at *https://www.nytimes.com/2018/06/21/opinion/facial-analysis-technology-bias.html*.

Buolamwini, J., and Gebru, T. (2018), "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification," in *Conference on Fairness, Accountability and Transparency, PMLR*, pp. 77–91, available at *http://proceedings.mlr.press/v81/buolamwini18a.html*.

Cairo, A. (2012), *The Functional Art: An Introduction to Information Graphics and Visualization*, Berkeley, CA: New Riders Publishing.

Carver, R., Everson, M., Gabrosek, J., Horton, N., Lock, R., Mocko, M., Rossman, A., Holmes, G., Belleman, P., Witmer, J., and Wood, B. (2016), "Guidelines for Assessment and Instruction in Statistics Education (GAISE) College Report 2016," available at *https://www.amstat.org/asa/education/Guidelines-for-Assessment-and-Instruction-in-Statistics-Education-Reports.aspx*.

Çetinkaya-Rundel, M., and Ellison, V. (2020), "A Fresh Look at Introductory Data Science," *Journal of Statistics Education*, 1–11, DOI: 10.1080/10691898.2020.1804497.

Chandra, R. V., and Varanasi, B. S. (2015), "Requests: HTTP for Humans," available at *https://2.python-requests.org/en/master/*.

Chu, J., Clinton, D., Dich, J., and Schonken, A. (2019), "COGS 108—Final Project—Crime Watch Effectiveness and Distribution," *GitHub*, available at *https://github.com/COGS108/FinalProjects-Fa19/blob/master/FinalProject-group030.ipynb*.

Donoho, D. (2017), "50 Years of Data Science," *Journal of Computational and Graphical Statistics*, 26, 745–766, DOI: 10.1080/10618600.2017.1384734.

Doucette, D. (2019), "Data Science Degrees: What Do Universities Need to Meet the Demand?," *Technology Solutions That Drive Education*, available at *https://edtechmagazine.com/higher/article/2019/01/data-science-degrees-what-do-universities-need-meet-demand-perfcon*.

Eubanks, V. (2018), *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*, New York: St. Martin's Press.

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., and Wenderoth, M. P. (2014), "Active Learning Increases Student Performance in Science, Engineering, and Mathematics," *Proceedings of the National Academy of Sciences of the United States of America*, 111, 8410–8415, DOI: 10.1073/pnas.1319030111.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020), "Array Programming With NumPy," *Nature*, 585, 357–362, DOI: 10.1038/s41586-020-2649-2.

Hicks, M. (2017), *Programmed Inequality: How Britain Discarded Women Technologists and Lost Its Edge in Computing*, Cambridge, MA: The MIT Press.

Hicks, S. C., and Irizarry, R. A. (2018), "A Guide to Teaching Data Science," *The American Statistician*, 72, 382–391, DOI: 10.1080/00031305.2017.1356747.

Hunter, J. D. (2007), "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, 9, 90–95, DOI: 10.1109/MCSE.2007.55.

JupyterHub (2017), available at *https://github.com/jupyterhub/jupyterhub*.

Keyes, O., Hutson, J., and Durbin, M. (2019), "A Mulching Proposal: Analysing and Improving an Algorithmic System for Turning the Elderly into High-Nutrient Slurry," in *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, CHI EA '19*, Association for Computing Machinery, New York, NY, USA, pp. 1–11, DOI: 10.1145/3290607.3310433.

Kluyver, T., Ragan-Kelley, B., Perez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., and Jupyter Development Team (2016), "Jupyter Notebooks—A Publishing Format for Reproducible Computational Workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, eds. F. Loizides and B. Scmidt, Amsterdam: IOS Press, pp. 87–90, DOI: 10.3233/978-1-61499-649-1-87.

Knaflic, C. N. (2015), *Storytelling With Data: A Data Visualization Guide for Business Professionals*, Hoboken, NJ: Wiley.

Kramer, A. D. I., Guillory, J. E., and Hancock, J. T. (2014), "Experimental Evidence of Massive-Scale Emotional Contagion Through Social Networks," *Proceedings of the National Academy of Sciences of the United States of America*, 111, 8788–8790, DOI: 10.1073/pnas.1320040111.

Krzywinski, M., and Cairo, A. (2013), "Storytelling," *Nature Methods*, 10, 687–687, DOI: 10.1038/nmeth.2571.

Lohr, S. (2012), "Opinion | Big Data's Impact in the World," *The New York Times*, available at *https://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html*.

Loper, E., and Bird, S. (2002), "NLTK: The Natural Language Toolkit," in *ETMTNLP '02: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, DOI: 10.3115/1118108.1118117.

Loy, A., Kuiper, S., and Chihara, L. (2019), "Supporting Data Science in the Statistics Curriculum," *Journal of Statistics Education*, 27, 2–11, DOI: 10.1080/10691898.2018.1564638.

Lue, R. A. (2019), "Data Science as a Foundation for Inclusive Learning," *Harvard Data Science Review*, 1, DOI: 10.1162/99608f92.c9267215.

Mangalindan, J. (2014), "How Tech Companies Compare in Employee Diversity," *Fortune*, available at *https://fortune.com/2014/08/29/how-tech-companies-compare-in-employee-diversity/*.

Markow, W., Braganza, S., Taska, B., Miller, Steven, M., and Hughes, D. (2017), "The Quant Crunch: How the Demand for Data Science Skills Is Disrupting the Job Market," available at *https://www.bhef.com/publications/quant-crunch-how-demand-data-science-skills-disrupting-job-market*.

Mason, H., and Patil, D. (2015), *Data Driven*, Sebastopol, CA: O'Reilly Media, Inc.

McKinney, W. (2010), "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, DOI: 10.25080/Majora-92bf1922-00a.

National Academies of Sciences, Engineering and Medicine (2018), *Data Science for Undergraduates: Opportunities and Options*, Washington, DC: National Academies Press, DOI: 10.17226/25104.

Nelson, B. (2014), "The Data on Diversity," *Communications of the ACM*, 57, 86–95, DOI: 10.1145/2597886.

Neumann, D. L., Hood, M., and Neumann, M. M. (2013), "Using Real-Life Data When Teaching Statistics: Student Perceptions of This Strategy in an Introductory Statistics Course," *Statistics Education Research Journal*, 12, 59–70.

Noble, S. U. (2018), *Algorithms of Oppression: How Search Engines Reinforce Racism*, New York: NYU Press.

Nolan, D., and Perrett, J. (2016), "Teaching and Learning Data Visualization: Ideas and Assignments," *The American Statistician*, 70, 260–269, DOI: 10.1080/00031305.2015.1123651.

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

O'Neil, C. (2016), *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*, New York: Crown Publishers.

Office for Civil Rights (2012), "Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule," *HHS.gov*, available at *https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html*.

Patil, P., Peng, R. D., and Leek, J. T. (2016), "A Statistical Definition for Reproducibility and Replicability," bioRxiv"no. 066803, DOI: 10.1101/066803.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011), "Scikit-Learn: Machine Learning in Python," *Journal of Machine Learning Research*, 12, 2825–2830, DOI: 10.5555/1953048.2078195.

Peters, B. (2012), "The Age of Big Data," *Forbes*, available at *https://www.forbes.com/sites/bradpeters/2012/07/12/the-age-of-big-data/*.

Piatetsky, G., and Gandhi, P. (2018), "How Many Data Scientists Are There and Is There a Shortage?," *KDnuggets*, available at *https://www.kdnuggets.com/2018/09/how-many-data-scientists-are-there.html*.

Porter, L., Bouvier, D., Cutts, Q., Grissom, S., Lee, C., McCartney, R., Zingaro, D., and Simon, B. (2016), "A Multi-institutional Study of Peer Instruction in Introductory Computing," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education, SIGCSE '16*, ACM, New York, NY, USA, pp. 358–363, DOI: 10.1145/2839509.2844642.

Prince, M. (2004), "Does Active Learning Work? A Review of the Research," *Journal of Engineering Education*, 93, 223–231, DOI: 10.1002/j.2168-9830.2004.tb00809.x.

Project Jupyter, Blank, D., Bourgin, D., Brown, A., Bussonnier, M., Frederic, J., Granger, B., Griffiths, T., Hamrick, J., Kelley, K., Pacer, M., Page, L., Pérez, F., Ragan-Kelley, B., Suchow, J., and Willing, C. (2019), "nbgrader: A Tool for Creating and Grading Assignments in the Jupyter Notebook," *Journal of Open Source Education*, 2, 32, DOI: 10.21105/jose.00032.

Raji, I. D., Gebru, T., Mitchell, M., Buolamwini, J., Lee, J., and Denton, E. (2020), "Saving Face: Investigating the Ethical Concerns of Facial Recognition Auditing," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, AIES '20*, Association for Computing Machinery, New York, NY, USA, pp. 145–151, DOI: 10.1145/3375627.3375820.

Richardson, L. (2007), "Beautiful Soup Documentation," available at *https://www.crummy.com/software/BeautifulSoup/bs4/doc/*.

Rivers, E. (2017), *Women, Minorities, and Persons With Disabilities in Science and Engineering*, Alexandria, VA: National Science Foundation, available at *https://ncses.nsf.gov/pubs/nsf19304/digest*.

Rubin, M. J. (2013), "The Effectiveness of Live-Coding to Teach Introductory Programming," in *Proceeding of the 44th ACM Technical Symposium on Computer Science education—SIGCSE '13*, ACM Press, Denver, CO, USA, p. 651, DOI: 10.1145/2445196.2445388.

Ruiz Junco, P. (2017), "Data Scientist Personas: What Skills Do They Have and How Much Do They Make?," *Glassdoor Economic Research*, available at *https://www.glassdoor.com/research/data-scientist-personas/*.

Russell, M. (2018), "University Students Flock to Data Science as Interest and Demand Surge," *Center for Digital Education*, available at *https://www.govtech.com/education/higher-ed/University-Students-Flock-to-Data-Science-as-Interest-and-Demand-Surge.html*.

Salian, I. (2017), "Universities Rush to Add Data Science Majors as Demand Explodes—SFChronicle.com," *San Francisco Chronicle*, available at *https://www.sfchronicle.com/business/article/Universities-rush-to-add-data-science-majors-as-12170047.php*.

Saltz, J. S., Dewar, N. I., and Heckman, R. (2018), "Key Concepts for a Data Science Ethics Curriculum," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education, SIGCSE '18*, Association for Computing Machinery, Baltimore, MD, USA, pp. 952–957, DOI: 10.1145/3159450.3159483.

Schlegel, K., Linden, A., Sallam, R., Howson, C., Sicular, S., Hare, J., Krensky, P., Tapadinhas, J., and Heizenberg, J. (2016), "Predicts 2017: Analytics Strategy and Technology," available at *https://www.gartner.com/en/documents/3531618/predicts-2017-analytics-strategy-and-technology*.

Seabold, S., and Perktold, J. (2010), "Statsmodels: Econometric and Statistical Modeling with Python," in *Proceedings of the 9th Python in Science Conference*, Austin, TX, pp. 92–96, DOI: 10.25080/Majora-92bf1922-011.

Shields, M. (2005), "Information Literacy, Statistical Literacy, Data Literacy," *IASSIST Quarterly*, 28, 6, DOI: 10.29173/iq790.

Sweeney, L. (2015), "Only You, Your Doctor, and Many Others May Know," *Technology Science*, 2015092903, *https://techscience.org/a/2015092903/*.

Taschuk, M., and Wilson, G. (2017), "Ten Simple Rules for Making Research Software More Robust," *PLOS Computational Biology*, 13, e1005412, DOI: 10.1371/journal.pcbi.1005412.

Tate, E. (2017), "Data Analytics Programs Taking Off at Colleges," *Inside Higher Ed*, available at *https://www.insidehighered.com/digital-learning/article/2017/03/15/data-analytics-programs-taking-colleges*.

Tufte, E. R. (1986), *The Visual Display of Quantitative Information*, Cheshire, CT: Graphics Press.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., and van Mulbregt, P. (2020), "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, 17, 261–272, DOI: 10.1038/s41592-019-0686-2.

Voytek, B. (2016), "The Virtuous Cycle of a Data Ecosystem," *PLOS Computational Biology*, 12, e1005037, DOI: 10.1371/journal.pcbi.1005037.

——— (2017), "Social Media, Open Science, and Data Science Are Inextricably Linked," *Neuron*, 96, 1219–1222, DOI: 10.1016/j.neuron.2017.11.015.

Waskom, M., Botvinnik, O., Ostblom, J., Gelbart, M., Lukauskas, S., Hobson, P., Gemperline, D. C., Augspurger, T., Halchenko, Y., Cole, J. B., Warmenhoven, J., de Ruiter, J., Pye, C., Hoyer, S., Vanderplas, J., Villalba, S., Kunter, G., Quintero, E., Bachant, P., Martin, M., Meyer, K., Swain, C., Miles, A., Brunner, T., O'Kane, D., Yarkoni, T., Williams, M. L., Evans, C., Fitzgerald, C., and Brian (2020), "mwaskom/seaborn: v0.10.1 (April 2020)," *Zenodo*, DOI: 10.5281/zenodo.3767070.

Wessel, J. R., Gorgolewski, K. J., and Bellec, P. (2019), "Switching Software in Science: Motivations, Challenges, and Solutions," *Trends in Cognitive Sciences*, 23, 265–267, DOI: 10.1016/j.tics.2019.01.004.

Wickham, H. (2014), "Tidy Data," *Journal of Statistical Software*, 59, 1–23, DOI: 10.18637/jss.v059.i10.

Wilkinson, L. (1999), *The Grammar of Graphics*, Berlin, Heidelberg: Springer-Verlag.

Williams, D. L., Beard, J. D., and Rymer, J. (1991), "Team Projects: Achieving Their Full Potential," *Journal of Marketing Education*, 13, 45–53, DOI: 10.1177/027347539101300208.

Wilson, G., Aruliah, D. A., Brown, C. T., Hong, N. P. C., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K. D., Mitchell, I. M., Plumbley, M. D., Waugh, B., White, E. P., and Wilson, P. (2014), "Best Practices for Scientific Computing," *PLoS Biology*, 12, e1001745, DOI: 10.1371/journal.pbio.1001745.

Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., and Teal, T. K. (2017), "Good enough practices in scientific computing," *PLOS Computational Biology*, 13, e1005510, DOI: 10.1371/journal.pcbi.1005510.

Wood, B. L., Mocko, M., Everson, M., Horton, N. J., and Velleman, P. (2018), "Updated Guidelines, Updated Curriculum: The GAISE College Report and Introductory Statistics for the Modern Student," *CHANCE*, 31, 53–59, DOI: 10.1080/09332480.2018.1467642.

Wu, X., and Zhang, X. (2017), "Automated Inference on Criminality Using Face Images," arXiv no. 1611.04135.

Wu, X., Zhu, X., Wu, G.-Q., and Ding, W. (2014), "Data Mining With Big Data," *IEEE Transactions on Knowledge and Data Engineering*, 26, 97–107, DOI: 10.1109/TKDE.2013.109.

Xie, Y., Allaire, J. J., and Grolemund, G. (2018), *R Markdown: The Definitive Guide*, Boca Raton, FL: Chapman and Hall/CRC.

Yan, D., and Davis, G. E. (2019), "A First Course in Data Science," *Journal of Statistics Education*, 27, 99–109, DOI: 10.1080/10691898.2019.1623136.

Yau, N. (2013), *Data Points: Visualization That Means Something*, New York: Wiley.

Zang, J., Dummit, K., Graves, J., Lisker, P., and Sweeney, L. (2015), "Who Knows What About Me? A Survey of Behind the Scenes Personal Data Sharing to Third Parties by Mobile Apps," *Technology Science*, 2015103001, available at *https://techscience.org/a/2015103001/*.