SUPPORTING DATA-INTENSIVE ENVIRONMENTAL SCIENCE RESEARCH: DATA SCIENCE SKILLS FOR SCIENTIFIC PRACTITIONERS OF STATISTICS

by

Allison Shay Theobold

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Statistics

MONTANA STATE UNIVERSITY
Bozeman, Montana

April 2020

DEDICATION

I would like to dedicate this dissertation to my partner, Laura Marie Smith. Laughably, inspiration for this research came from an argument about Ornate box turtles, en route to ski. That day and every day after, you've reminded me the power of empathy and have inspired me to be courageous and make a difference. I could not have gotten through this stage in life without you and your support. Even with a global pandemic surrounding us, you've made each and every milestone feel incredibly special. Your thoughtfulness and compassion is an inspiration to me and everyone around you. I love you to the moon and back.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my adviser, Dr. Stacey Hancock, for her unwavering support and belief in me. I cannot thank Stacey enough for the countless hours she spent reviewing my writing, our thoughtful discussions of future directions for my research, and always pushing me to be my best. I also owe gratitude to my committee members: Dr. Jenny Green, Dr. Megan Wickstrom, Dr. Mary Alice Carlson, and Dr. Mark Greenwood. You have each challenged and supported me growing as a researcher and an educator, with seemingly endless encouragement—a role I hope to someday fill with my students.

I want to thank my family for the support and strength they've given me throughout my degree. Reford—you've always pushed me to be my best, to overcome obstacles, and to not be afraid to speak up. Karen—the kindness you give everyone is inspiring, you have shown us how to live the golden rule every day. Kalinda—you are simultaneously a pillar of strength and the most brilliant ray of sunshine, you make me feel like the luckiest sister alive. Marilyn—you have shown me that life has no bounds, that love is unconditional, and that there is no stronger bond than family.

Finally, I would like to thank the strong women that I have had the privilege to learn from. Barbara Milburn was the first educator to show me what it means to teach *every* student. Dr. Tracii Friedman taught me the value of struggling in mathematics, and gave me the the strength to not walk away. In every class, I strive to have the tenacity with which these women approach teaching.

VITA

Allison Shay Theobold was born in Grand Junction, Colorado to Karen and Reford Theobold. After graduating from Grand Junction High School in 2007, Allison attended Colorado Mesa University. In 2014 Allison graduated cum laude with a bachelors in Mathematics, with a concentration in Statistics, and a bachelors in Business Administration, with a concentration in Economics.

In 2014 Allison moved to Bozeman, Montana to pursue a doctoral degree in Statistics from Montana State University. During this time, Allison served as the instructor of record for seven sections of Introductory Statistics and Intermediate Statistics courses, teaching over 225 undergraduate students. Allison also served as a statistical consultant to Montana State University students, faculty, and staff at the Department of Mathematical Science's Statistical Consulting and Research Services (SCRS). As part of her dissertation, Allison created and taught a series of data science literacy workshops. Through partnerships with the Montana State Library and SCRS, these workshops will continue to serve Montana State's broader research community for years to come.

Allison's dedication and success in teaching were acknowledged by the college and department by the receipt of the College of Letters and Sciences Outstanding Graduate Teaching Assistant Award and the Department of Mathematical Sciences Outstanding Graduate Teaching Assistant Award. The importance and impact of Allison's research has been recognized by her receipt of the Kopriva Graduate Student Fellowship, Honorable Mention Speed Session Award from the Statistics and Data Science Education Section of the ASA, and the Department of Mathematical Science's Gary Sackett Research Fellowship.

Upon receiving her degree, Allison moves forward as a statistics educator, excited to create statistics and data science classrooms that promote diversity and inclusion through foundational understandings of how data impact our everyday lives. Allison will begin her next chapter in the fall of 2020 as an Assistant Professor of Statistics at California Polytechnic University in San Luis Obispo, California.

TABLE OF CONTENTS

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

LIST OF TABLES

LIST OF TABLES – CONTINUED

LIST OF FIGURES

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

ABSTRACT

The importance of data science skills for modern environmental science research cannot be understated, but graduate students in these fields typically lack these integral skills. Yet, over the last 20 years statistics preparation in these fields has grown to be considered vital, and statistics coursework has been readily incorporated into graduate programs. As "data science" is the study of extracting value from data, the field shares a great deal of conceptual overlap with the field of Statistics. Thus, many environmental science degree programs expect students to acquire these data science skills in an applied statistics course. A gap exists, however, between the data science skills required for students' participation in the entire data analysis cycle as applied to independent research, and those taught in statistics service courses. Over the last ten years, environmental science and statistics educators have outlined the shape of the data science skills specific to research in their respective disciplines. Disappointingly, however, both sides of these conversations have ignored the area at the intersection of these fields, specifically the data science skills necessary for environmental science *practitioners* of statistics.

This research focuses on describing the nature of environmental science graduate students' need for data science skills when engaging in the data analysis cycle, through the voice of the students. In this work, we present three qualitative studies, each investigating a different aspect of this need. First, we present a study describing environmental science students' experiences acquiring the computing skills necessary to implement statistics in their research. In-depth interviews revealed three themes in these students' paths toward computational knowledge acquisition: use of peer support, seeking out a "singular consultant," and learning through independent research. Motivated by the need for extracurricular opportunities for acquiring data science skills, next we describe research investigating the design and implementation of a suite of data science workshops for environmental science graduate students. These workshops fill a critical hole in the environmental science and statistics curricula, providing students with the skills necessary to retrieve, view, wrangle, visualize, and analyze their data. Finally, we conclude with research that works toward identifying key data science skills necessary for environmental science graduate students as they engage in the data analysis cycle.

1

INTRODUCTION

Over the last two decades, nearly every scientific field has seen a rapid increase in the volume and variety of available data and a growth in the usage and power of computational tools to model phenomena. This increased focus on data-intensive research has made computationally heavy applications of data science techniques—such as management and coalition of large data sets, high dimensional data visualization, and Bayesian modeling—essential understandings for scientific research. These dramatic changes to scientific practices have created a crucial need to reevaluate how our educational system can better prepare current and future generations of researchers (Green et al., 2005; Hampton et al., 2017). Unfortunately, the gap between the computing included in the education students receive and the computational knowledge required for scientific research has become more pronounced, especially in the environmental and life sciences. When considering the issue of curriculum reevaluation, we note that over the last 20 years, statistics preparation in these fields has become vital.

The History of Computing in Statistics and the Environmental Sciences

The Emergence of Computing in the Statistics Curriculum

In 1962, John Tukey charged the field of Statistics to "seek out novelty in data analysis," reflecting that "in the future [Statistics] can and should contribute much more" to data analysis (Tukey, 1962, p. 3). This charge for innovation in data analysis was echoed in Breiman's organization of the "Conference on the Analysis of Large Complex Data Sets" in 1977 and the symposium on "Modern Interdisciplinary

University Statistics Education" by the Committee on Applied and Theoretical Statistics' (CATS) in 1992. In its August 1992 meeting in Boston, CATS "noted widespread sentiment in the statistical community that upper-level undergraduate and graduate curricula for statistics majors are currently structured in ways that do not provide sufficient exposure to modern statistical analysis and computational and graphical tools." This growth the field of Statistics had experienced "is not reflected in the education that future statisticians receive," and left the need for a more meaningful integration of the "computational and graphical tools that are today so important to many professional statisticians" (National Research Council, 1994, p. vii).

At the close of the century, this call for the need to transform the undergraduate statistics curriculum was reiterated by statistics educators, mapping opportunities for innovation to bring statistics education up to speed with the modern day practice of Statistics. Moore et al. (1995) speculated that "technological advances may at last bring widespread change to college teaching" (p. 250), imagining plausible futures for statistics teaching at the university level. Biehler (1997) added to these musings, elaborating on the need for educators to "critically evaluate existing software and to produce future software more adequate both for learning and doing statistics in introductory courses" (p. 167). Possibilities for modernizing the outdated and overly mathematical undergraduate statistics programs were voiced by Higgins (1999), with Nolan and Speed (1999, 2000) providing recommendations for infusing computing explorations into "traditional" mathematical statistics course(s).

The next century brought the introduction of "data science" (Cleveland, 2001), and continued conversations surrounding how to infuse computing into the statistics curriculum. However, the majority of these conversations revolved around including computing into mathematical statistics course(s) (Reid et al., 2003; Horton

et al., 2004). Despite these small changes, some statisticians remained concerned with the trajectory of the field of Statistics. Friedman, reflecting on the absence of Statistics from the development and implementation of data mining methodology, lamented that "the field [of Statistics] should be defined in terms of a set of *problems* rather than a set of tools, namely those that pertain to data" (Friedman, 2001, p. 8, emphasis in original). Furthermore, the field needed to "make peace with computing," because it had been "one of the most glaring omissions in the set of tools that have so far defined Statistics" (Friedman, 2001, p. 8). That same year, Breiman, the creator of classification and regression trees, published a groundbreaking piece on the two cultures of statistical modeling, data modeling and algorithmic modeling. These cultures, Breiman argued, use two fundamentally different methods to model the relationship between two sets of data: the inputs to a process or mechanism and the outputs from that process. Data modeling assumes that the relationship between these two datasets can be explained by a mathematical model, but relies entirely upon the correct model. In contrast, algorithmic modeling determines a data model solely in terms of correctly predicting an output, provided an input, allowing for the model to potentially have little to no relationship with the underlying data-generating process. Breiman asserted that Statistics' commitment to data modeling had prevented the field from entering new arenas where "the data being gathered is not suitable for analysis by data models" (p. 200). Hence, Breiman encouraged statisticians to become more familiar with algorithmic modeling, to address this significant change in the data landscape.

Although these calls for an increased focus on computational tools continued to be heard throughout the statistics community, it wasn't until nearly ten years later that Brown and Kass resumed the discussion around the statistical training of undergraduate and graduate Statistics majors. This work came at a critical time,

following Peck and Chance's detailed description of the assessment of Cal Poly's undergraduate Statistics program (2005). Brown and Kass argue that "to remain vibrant, the field [of Statistics] must open up by taking a less restrictive view of what constitutes statistical training" (2009, p. 105). The authors acknowledged that "a fear lurks in the heart of many statistics professors" where "statistics as we know it [may] become obsolete" if the field continued to complacently ignore the innovation in data analysis techniques (p. 105). They found that, while programs emphasize the mathematical logic of data analysis, when faced with an actual data analysis, "graduate students in statistics often are reticent to the point of inaction" (p. 106).

At the climax of these discussions, came the publication of "Computing in the Statistics Curriculum" from Deb Nolan and Duncan Temple Lang (2010). In this influential article, Nolan and Temple Lang painted a broad picture of the computing skills that successful statisticians must be facile with, and how these skills had been infused into the Statistics program at the University of California, Berkeley. The authors asserted that, "the skill set needed by a statistician even 20 years ago is very different from what is needed today (p. 98). Moreover, as a statistics education community, we were not preparing students with the computational proficiency, the statistical problem solving, or the "confidence needed to overcome computational challenges" (p. 97). Nolan and Temple Lang reflected that they have "found that Bachelors and Masters students who enter the workforce spend much of their efforts retrieving, filtering, and cleaning data and doing initial exploratory data analysis," (p. 99), but students were not taught these skills in their courses. Instead, students were "told to learn how to program by themselves, from each other, or from their teaching assistant in a two-week 'crash course' in basic syntax at the start of a course" (p. 100). They outlined a series of recommendations for changes the statistics education community should make to bring the statistics curriculum up to date with the tools

that modern statisticians use, so that students would leave the statistics curriculum with the "computational understanding, skills, and confidence needed to actively and wholeheartedly participate in the computational arena" (p. 106).

The Emergence of Computing in the Environmental Science Curriculum

Meanwhile, in the 1990s and 2000s, the environmental science community was having similar conversations surrounding the importance of computing for research in biological fields. In 1977, Levin et al. pioneered these conversations by asserting that, with the addition of more powerful computers and new analysis techniques, the "face of the science of computational population biology and ecosystems science will change in the next decade" (p. 341). This conversation went unanswered until in 2001, George Johnson of the New York times published a piece describing the use of computing for research in bioinformatics, concluding that no matter what scientific field one chooses to perform research in, "all science is computer science" (Johnson, 2001).

Finally, by the early 2000s, the conversation around teaching computing in environmental science courses began to flourish. Andelman and colleagues (2004) published a recount of an interdisciplinary seminar they developed for graduate students and what they learned about students' computational skills—or lack thereof. During their course, the authors learned that students were unprepared with both the statistical and computational skills necessary for data analysis; rather, "ninety-three percent of students did not have skills in the scripted programming languages (e.g., SAS or MATLAB) that are needed for the integration of large data sets" (p. 244). As a consequence, "the greatest limitation [...] that the students faced was related to data concatenation, manipulation, and analysis" (p. 245). Environmental science educators continued to press on the issue of integrating computing into environmental

science research (Green et al., 2005; Hastings et al., 2005), detailing the formidable computational challenges scientists were facing. To stress the importance of these needs, the NSF sponsored a series of three workshops on "quantitative environmental and integrative biology," to aid in identifying "areas of cutting edge research in ecology and environmental biology that require integration with novel computational, statistical, and informatics tools" (Green et al., 2005, p. 502).

The importance of every scientific researcher having the ability to reason through computational problems, was further emphasized in 2006 by Jannette Wing in an Association for Computing Machinery (ACM) communication. Wing pioneered the concept of computational thinking in her ACM communication, declaring computational thinking to be a "fundamental skill for everyone" (p. 33). Wing then outlined the variety of mental tools that encompass computational thinking, including but not limited to: thinking recursively, parallel processing, abstraction, and decomposition. A series of articles from the ITiCSE and SIGCSE conferences followed, with computer science educators describing their institution's development of introductory computer science courses for non-computer science majors. These courses were designed for "any student intending to major in science or engineering" (Dodds et al., 2007, p. 23) and were intended to develop students' problem solving and programming skills, paint a compelling picture of the vastness of computer science, and attract students to continue to study computer science (Dodds et al., 2007, 2008; Hambrusch et al., 2009; Wilson et al., 2008).

During these conversations, Greg Wilson and Dr. Brent Gorda at the University of Toronto developed a course named Software Carpentry, to teach "scientists and engineers the 'common core' of modern software development" (Wilson, 2006, p. 66). Software Carpentry addressed a gaping hole in graduate education in the sciences, providing students with the tools to increase their productivity by improving the

quality of their code. This course included topics such as version control, scripting, debugging, testing, and continuous integration, all topics which few, if any, students had seen before.

The conversation around scientists' need to be familiar with scripted programming and reproducible documents continued with Stephen Eglen's tutorial piece on how to teach `R` programming to computational biology students (Eglen, 2009). (It is worth noting that Eglen's piece is potentially the first occurrence in which the use of `R` is advocated for research in the biological sciences.) Rounding out the decade, Kelling and colleagues revisited the original argument of "data-intensive" science in 2003, reiterating the importance of computing to biological research, and outlining the big picture "steps in the data-intensive science workflow" (Kelling et al., 2009, p. 614). These researchers paid special attention to the integration of statistics in this data-intensive workflow, with exploratory analysis and confirmatory analysis encompassing the final two stages. The authors concluded with a charge for environmental scientists to "overcome the challenges in organizing and analyzing massive and heterogeneous data" so the field could make headway towards unraveling the complexity of ecological systems" (p. 619).

Data Science in the Environmental Sciences

With the importance of computing to environmental science fields firmly in place, the literature over the next decade focused instead on the important role academic institutions have in preparing undergraduate and graduate students with the skills relevant to research in these fields. Strasser and Hampton (2012) began this conversation, focusing on the importance of data management in undergraduate ecology courses. The authors reported that while ecology instructors rated data management topics, such as workflows, databases, and reproducibility, as very

important for their research, less than 20% of instructors included these topics in their courses. These results suggested that—across institutions—"data management education is not currently a priority for ecology instructors" (p. 10).

That same year, Hernandez, an environmental science graduate student, led a large scale study of the "technological and computational experiences of environmental scientists in the formative stages of their career" (Hernandez et al., 2012, p. 1068). These researchers found that over 74% of the students surveyed stated they had no skills in any programming language—including `R`—and only 17% reported basic skill levels in any programming language. These findings suggested that—across institutions—graduate students were not obtaining the knowledge and skills required to navigate the advancing fields of technology, computation, and data management through their coursework or instruction.

Given the poor computational preparation of environmental science graduate students by their curriculum, Hernandez et al. suggested that student-focused workshops could "provide intensive environments" where students could learn "particular methods or technologies" (p. 1075). Furthermore, developing and offering these workshops would be simpler than developing new courses to organize and implement. Over the subsequent years, researchers would reiterate the ability of these external workshops to provide students with on-demand, intensive training to acquire the computing skills necessary for data-intensive environmental science research.

Gutlerner and Van Vactor (2013) led the charge in the development of short-format, skill-building courses, in an article describing tools for evolving the scientific curriculum. These short-courses allowed for students to "take a course on a particular topic or technique at the time when they are most motivated to learn about it" (p. 732). Alternatively, these intensive experiences could be harnessed into a "bootcamp" prior to students' first semester of graduate school. One such "quantitative methods

bootcamp" was implemented by instructors at Harvard Medical School, to "enable students to use computational tools to visualize and analyze data" and to "strengthen their computational thinking skills" (Stefan et al., 2015, p. 1). The authors argued introducing graduate students to these concepts before they begin their coursework lowered the computational barrier for students before taking courses, empowered students to learn computational tools on their own, and enabled courses to "build upon this foundation and integrate quantitative methods throughout the curriculum" (p. 2).

Around the same time, Greg Wilson created the Software Carpentry Foundation, transforming the Software Carpentry course into a workshop curriculum, which could be offered to researchers around the world. Software Carpentry found larger success than its predecessors, due in part to the dramatic change in the scientific landscape. Backed by the support of the Mozilla and Sloan Foundations, in 2013, Software Carpentry offered its first Software Carpentry workshops for Librarians in the United States and Canada. Then, in 2014, Data Carpentry was founded to "train researchers in the core data skills for efficient, shareable, and reproducible research practices" (2020), specific to their field of research. These workshops met the need for "good training resources for researchers looking to develop skills that will enable them to be more effective and productive" (Teal et al., 2015, p. 135). Teal and collaborators pressed further into the findings of Hernandez et al. (2012) and Wilson (2016), claiming that "most or all of what [researchers] know about data management, analysis, and sharing has been learned piecemeal, or not learned at all," as "training in data and computing skills is still largely absent from undergraduate and graduate programs" (p. 136). The authors emphasized the importance of developing streamlined training opportunities for researchers in these fields for two main reasons: (1) there is substantial variation in computational training at every institution, and

(2) for students not being directly taught computing skills, it is difficult to wade through the plethora of online lessons, MOOCs, and books to find relevant resources. The authors acknowledged that, while the Data Carpentry workshops "will not be able to teach researchers all of the skills they need in two days," the workshops "are a way to get started," lowering the activation energy required and empowering researchers "to be able to conduct the analyses necessary for their work in an effective and reproducible way" (p. 143).

Culminating all of these conversations, in 2017, educators from a variety of environmental science research areas gathered together to write a formative piece on the skills and knowledge necessary for "data-intensive" environmental science research (Hampton et al., 2017). Hampton and colleagues outlined the current state of environmental science education in American universities, stating that "a symptom of the current curriculum's shortcomings is the recent emergence of a variety of extramural options for acquiring critical technological skills" (p. 547). They then describe the "skillset required by environmental scientists to succeed in the kind of data-intensive scientific collaboration that is increasingly valued" (p. 548). These five classes of skills included: (1) data management and processing, (2) analysis, (3) software skills for science, (4) visualization, and (5) communication methods for collaboration and dissemination. Each of these classes of skills reiterates previous research on the "good enough" (Wilson et al., 2017) computational skills necessary for research these fields.

Over the last 20 years, however, attention has yet to be paid to the substantial role students' statistics education potentially plays in their attainment of the data science skills necessary for data-intensive scientific research. Andelman and colleagues reflected that, in addition to students' lack of familiarity with scripted programming languages, students were also "unfamiliar with multivariate statistics and with the

range of models for regression and analysis of variance" (p. 244). Almost 10 years later, Strasser and Hampton reported that the most common courses ecology faculty voiced as possibly covering "data-related topics" were "ecology laboratories, advanced ecology courses, or statistics courses" (p. 7-8). The survey administered by Hernandez et al. asked students whether they had taken or planned to take courses related "to the management and analysis of large or complex data" (p. 1070), including courses in spatial or time series analysis. The authors also surveyed students regarding their level of proficiency with programming languages, including `R`. Consistent with the previous discussions, Hampton et al. continued to outline the extensive statistical skills a data-intensive environmental science researcher should possess, but do not admit that today, the majority of students in these fields complete statistics coursework prior to graduation. This discontinuity in these conversations comes as a shock, as, over this time period, "statistical preparation in the environmental sciences has grown to be considered vital" (Hampton et al., 2017, p. 547).

Data Science in Statistics

In statistics education, during the 2010s the research focused on integrating computing throughout the statistics curriculum, revising the program expectations for undergraduate statistics programs, and creating user-friendly tools for streamlined data science workflows. In the year following the publication of "Computing in the Statistics Curriculum" (Nolan and Temple Lang, 2010), the Mckinsey Report (Manyika et al., 2011) was published. The McKinsey report stated that, by 2018, "the United States alone could face a shortage of 140,000 to 190,000 people with deep analytical skills as well as 1.5 million managers and analysts with the know-how to use the analysis of big data to make effective decisions" (p. 3). Simultaneously, during the 2011 United States Conference on Teaching Statistics (USCOTS), statistics

educators began conversations around how computing could play a larger role in the introductory statistics course through the incorporation of simulation-based methods. Following these conversations, two simulation-based introductory statistics textbooks emerged, both carrying with them a suite of applets for student use (Lock et al., 2013; Rossman and Chance, 2011).

Amidst these conversations, a suite of packages was being created, which would fundamentally change how users interact with `R`. The `ggplot2` `R` package, created by Hadley Wickham in 2005, spearheaded the change toward creating user-friendly `R` tools all "sharing an underlying design philosophy, grammar, and data structures" (Wickham, 2017). The `ggplot2` package was created to produce statistical, or data, graphics; but, unlike most other graphics packages, it had the "deep underlying grammar" (Wickham, 2016, p. 1) of Wilkinson's Grammar of Graphics (2005). Over the next decade, Wickham and his team produced the suite of packages now included in the `tidyverse` package, namely `stringr` (2009), `dplyr` (2014), `RMarkdown` (2014), `tidyr` (2014), `readr` (2015), `purrr` (2015), `tibble` (2016), and `forcats` (2016). The `tidyverse` package houses all of the necessary packages to import, tidy, transform, wrangle, visualize, and model data, and to communicate the results.

With calls for transforming undergraduate statistics education resounding nationally, the 2014 American Statistical Association (ASA) President, Nathaniel Schenker, convened a workgroup to update the association's guidelines for undergraduate programs. The group, with broad representation from academia, industry, and government, put forward guidelines that were endorsed by the ASA Board of Directors in November 2014 (American Statistical Association Undergraduate Guidelines Workgroup, 2014). These new guidelines included an increased emphasis on data science skills and real applications, specifically students' ability to "access and

manipulate data in various ways, use a variety of computational approaches to extract meaning from data, [and] program in higher-level languages" (p. 7).

Although these changes reflected a growing consensus that computing should be featured throughout statistics programs, much of the statistics education literature up to that point had focused on the introductory statistics and mathematical statistics courses. Hence, in 2015, *The American Statistician* produced a special issue on "Statistics and the Undergraduate Curriculum," to encourage submissions of broader topics in the statistics curriculum. The articles in the special issue fell primarily into two themes: the first theme described how computing should be included throughout the statistics curriculum, with articles from Green and Blankenship (2015), Tintle and colleagues (2015), and Hesterberg (2015); the second theme in these articles presented thoughts on how data science topics should be integrated into undergraduate statistics courses, with articles from Nolan and Temple Lang (2015), Grimshaw (2015), Baumer (2015), and Hardin et al. (2015). In the same issue, George Cobb provocatively stated that the statistics curriculum needed to be rebuilt "from the ground up" (2015), as "what we teach lags decades behind what we practice" and "the gap between our half-century-old curriculum and our contemporary statistical practice continues to widen" (p. 268). In his article, Cobb argued that statistics, like computer science, should be teaching algorithmic thinking at a basic level. But, computing should be mindfully included throughout the statistics curriculum, rather than simply inserting "a new computing course into the existing curriculum" (p. 275).

Despite these technological advances promoting a facile integration of data science in the statistics curriculum and calls for purposeful inclusion of computing in the statistics curriculum, we continue to see students from scientific disciplines leave the statistics classroom without data science skills in hand. A mere 60% of environmental science graduate students reported a basic skill level in `R` (Hernandez

et al., 2012, p. 1069), which has become the "primary tool" reported for data analysis in environmental science research (Lai et al., 2019, p. 1). This gap between the importance of data science reiterated by statistics educators and the data science skills environmental science graduate students report leaving their program with demonstrates that data science concepts continue to be absent from many statistics courses. To promote conversations such as this, the *Journal of Statistics Education* will publish a special issue on "Computing in the Statistics Curriculum" in July 2020. To celebrate the 10-year anniversary of Nolan and Temple Lang's pioneering piece, articles in the special issue will look into what has changed since the publication of "Computing in the Statistics Curriculum," what still needs to change, and what is needed to implement curricular shifts.

Barriers to Incorporating Data Science in the Curriculum

While calls for incorporating computing throughout the environmental science and statistics curricula have resonated for the last ten years (Jones et al., 2006; Joppa et al., 2013; Laney et al., 2015; Manyika et al., 2011; Mokany et al., 2016; Peters and Okin, 2017; Smith, 2015; Teal et al., 2015), we continue to see researchers reporting the computational ill-preparation of environmental science undergraduate and graduate students by their curriculum (Hampton et al., 2017; Teal et al., 2015). This raises the question, why are these skills still so rare when the need for them is now widely recognized?

Nearly ten years ago, over 70% of ecology instructors reported substantial barriers to incorporating data management topics in their course(s). These barriers include: the instructor's lack of time or their lack of knowledge of the topics, students' lack of the necessary quantitative understandings, or a lack of alignment of the data management topics with the content of their course. These obstacles can be distilled

into two main components: first we are "attempting to fit more material into already-full courses and curriculum," and second, these courses are potentially "taught by people who do not feel prepared to address topics relevant to big data and data-intensive research" (Hampton et al., 2017, p. 547). Yet, this lack of computational training impedes the progression of scientific research and results in substantial hidden costs.

Instead of acquiring these necessary skills in the coursework required for their programs, these environmental science graduate students "learn much of what they know about programming and data management on their own or the information is passed down within a lab" (Teal et al., 2015, p. 136). Despite the inclusion of statistics courses in these students' programs of study, students continue to be "told to learn how to program by themselves, from each other, or from their teaching assistant in a two-week 'crash course' in basic syntax " at the beginning of their statistics course (Nolan and Temple Lang, 2010, p. 100). This teach-yourself approach sends the signal to students that "computing is not of intellectual importance relative to the material covered in lectures" (Nolan and Temple Lang, 2010, p. 100). Moreover, this structure results in students potentially "picking up bad habits, misunderstandings, and, more importantly, the wrong concepts" (Nolan and Temple Lang, 2010, p. 100). Students' initial knowledge shapes the methods they use to accomplish a task, making some tasks impossible. They may spend weeks or months doing things that could be done in hours or days, unable to abstract what they learned to broader classes of tasks. Furthermore, students may be unaware of the reliability and reproducibility of their results.

Specific Aims for This Research

Clearly, the current situation is unsatisfactory; however, few efforts have been made to better understand the data science skills necessary for environmental science graduate students as they implement statistics in their research. The findings of Hernandez et al. suggest that—by in large—graduate students are not acquiring the data science skills required for participation in data-intensive research in their curriculum. Yet, elements of these skills are necessary for each student as they engage in their research, which surfaces the question: how are environmental science graduate students acquiring the data science skills necessary to implement statistics in the context of their research? Investigating students' experiences navigating the phenomenon of acquiring the data science skills necessary for their research adds a new perspective to the conversation surrounding the acquisition of data science skills for scientific research, and brings to light the pathways through which students successfully acquire these necessary skills.

Multiple environmental science researchers reference extracurricular workshops as a potential solution for researchers acquiring data science skills 'just in time' for their research (Teal et al., 2015; Hampton et al., 2017). Namely, Data Carpentry workshops provide researchers with "high-quality, domain-specific training covering the full lifecycle of data-driven research" (2020). Although Data Carpentry workshops are developed by the community to be tailored to specific areas of research, such as Ecology, there has been no formal investigation on the relevance of the skills taught in these workshops to environmental science graduate students, a population of researchers in critical need for relevant, high quality, and accessible computing instruction. Understanding the data science skills relevant to this population of researchers allows for the tailoring of current workshop resources, by making evidence-

based, iterative improvements to the content and structure of the workshops.

These profound changes in the data landscape have also impacted the instructors of graduate courses which are intended to arm environmental science students with the data science skills necessary for their independent research. Instructors may be experiencing similar barriers as those faced ten years prior (Strasser and Hampton, 2012). Instructors of these courses may not "have not been taught computing formally," so they "have not had the opportunity to learn it well, and feel they cannot teach it effectively" (Nolan and Temple Lang, 2010, p. 106). Educators from both Statistics and the environmental sciences have outlined data science skills of potential relevance to researchers in their respective field, but each of these conversations neglect a critical aspect of data-intensive environmental science research, the data analysis cycle.

While we may see data science concepts integrated into the undergraduate programs in statistics, integrating these topics into graduate-level statistics service courses, often required for environmental science graduate students, has received less attention and poses different issues. These statistics courses that serve a variety of students reflect a snapshot of the statistics curriculum, but often act as students' sole statistics course prior to conducting the research required for their degree. Thus, instructors of these courses are forced to navigate difficult decisions of how they can ensure their students leave the classroom with both the statistical and "computational understanding, skills, and confidence needed to actively and wholeheartedly participate" in the scientific research arena (Nolan and Temple Lang, 2010, p. 106). Regrettably, for instructors unfamiliar with students' scientific disciplines, it can be difficult to "be bold" and infuse data science skills relevant to students' field of research into the classroom (Nolan and Temple Lang, 2010, p. 106).

Each of these issues facing environmental science students and faculty necessitates a better understanding of the specific data science skills relevant to environmental science graduate students as they engage in the data analysis cycle. Understanding the data science skills relevant to this population of researchers allows for the evaluation of the content included in tailored extracurricular workshops and provides statistics and environmental science educators with a set of foundational data science concepts to be included throughout the environmental science graduate curriculum.

With these considerations in mind, the goals of this research are threefold: (1) to outline the experiences of graduate students in the environmental sciences when acquiring the data science skills necessary to apply statistics in the context of their research, (2) to design, implement, and evaluate a suite of data science workshops tailored for graduate-level environmental science researchers, and (3) to describe the data science skills environmental science graduate students employ throughout their research when engaging in the data analysis cycle, and how these skills evolve over time.

For this research, the collection of fields who perform research in the biological and environmental science fields are captured under the term "environmental science." At Montana State University, these are the fields whose students are required or highly-recommended to enroll in the graduate-level Applied Statistics course sequence. In this research, I use the following terms interchangeably: "computing skills necessary to implement statistics," "statistical computing," and "data science skills." Each of these terms are considered to consist of the computing knowledge and skills necessary for the entire data analysis cycle, from data cleaning to data visualization to data analysis to communication. The computing skills necessary throughout the data analysis cycle may include general programming concepts such as

loops, user-defined functions, or conditional statements, but the focus of data science skills differ fundamentally from general programming skills. Rather than focusing on computer architecture, design, and application, for data science skills, data are the focus.

Research Journey

I was drawn to research in data science education through my experiences as a second-year graduate student in Statistics. During my second year, I provided statistical consulting for a graduate student in Ecology. This graduate student sought out consulting for assistance to implement a Bayesian framework to Ornate Box Turtle mark-recapture data, having no previous experiences working with statistical software. A component of our consulting collaboration took the form of weekly `R` workshops covering a variety of skills, from importing data to writing for-loops and functions, to fitting models in the `R` package `rjags`. At the close of the semester, I appreciated the computational challenges environmental science researchers face in their attempts to implement statistics in their research, and a realization of the data science skills with which graduate students typically leave their statistics courses. This emboldened me to investigate how environmental science graduate students acquire the data science skills necessary for research in their fields.

Experiences of Environmental Science Graduate Students

With this motivation in hand, I set out to design a study to understand and describe graduate students' transferability of the data science skills learned in the statistics classroom to environmental science applications. The design of this pilot study was comprised of two parts: (1) students' completion of hands-on computational problems, and (2) a survey of students' attitudes and experiences learning and using

computing skills.

The computing tasks were included to assess students' abilities to reason through applications of data science skills in an environmental science context. Then, after reasoning through each task, students were asked to detail where and how they had acquired the computational skill(s) they had employed while completing the task. During my initial data analysis, I realized that if a student was unable to reason through a particular task, that did not necessarily capture their ability to reason through that type of data science application in their field. Indeed, it is possible that the student was unable to reason through that type of data science task or, alternatively, the task in question may have been irrelevant to the "typical" data science applications in the student's respective field of research. Therefore, these data science problems were removed from the focus of this study.

Although the statistical computing tasks may not have accurately captured students' data science understandings, the interviews accompanying these tasks shed light on how students acquired the data science skills they were familiar with. In an article which appeared in the *Statistics Education Research Journal* (Theobold and Hancock, 2019), Chapter 2 outlines the results of these in-depth interviews, how these graduate students experienced the phenomenon of acquiring the computational skills necessary to implement statistics in their research. Three themes emerged in students' paths towards computational knowledge acquisition: use of peer support, seeking out a singular "consultant," and learning through independent research experiences. These themes provide descriptions of graduate student experiences absent from the environmental science literature, informing how instruction can be improved, both in and out of the formal classroom.

The findings of this phenomenological study led me to wonder how students' acquisition of the data science skills necessary for their research could be facilitated

with extracurricular workshops tailored to research in their specific field. Current ecology focused extracurricular workshops, such as Data Carpentry, aim to provide researchers with the fundamental data skills needed to conduct research in that field. However, the skills included in these workshops may not reflect the key data science skills necessary for the population of environmental science graduate student researchers. Therefore, this research demanded an understanding of the key data science skills necessary for environmental science graduate students to implement statistics in their research. These questions led to the two follow-up studies detailed in Chapters 3 and 4.

Designing Data Science Workshops for Data-Intensive Environmental Science Research

The first follow-up study focused on (1) describing the computing skills environmental science faculty believe are necessary when implementing statistics in graduate-level environmental science research, (2) investigating how these data skills can be infused into currently existing extracurricular data science workshops, and (3) understanding the backgrounds and experiences of attendees of these workshops.

For these investigations, we executed a three-phase design-based implementation research model (Fishman et al., 2013). Phase one encompassed conducting in-depth interviews with faculty members from environmental science fields regarding the computational skills they believed are necessary for graduate students to engage in the data analysis cycle in their research. Phase two then focused on adapting currently existing workshop resources to design a series of data science workshops targeting the key computational skills distilled from these faculty interviews. Phase three consisted of implementing the workshops and collecting survey responses from the workshop attendees regarding their backgrounds prior to the workshop and their experiences participating in each workshop.

For phase one, all university faculty currently overseeing a graduate student from the departments of Ecology, Land Resources and Environmental Sciences, Animal and Range Sciences, and Plant Sciences and Plant Pathology at Montana State University were emailed requesting their participation in this research. Faculty members from these fields were included because of the large degree of overlap in the type of data collected and analyzed in these fields. Therefore, graduate students from these fields would presumably have similar computational skills required of them as they analyze their data. A total of 61 faculty members were invited to participate in the study, and 23 total faculty agreed to participate in an interview.

During these interviews, faculty were asked a series of questions detailing the computational skills they believe are necessary for graduate students in their field to implement statistics in their research. Over the course of transcribing these interviews, it became clear to me that many faculty focused on the statistical skills and understandings necessary for graduate students to succeed in their research, rather than the computing skills necessary to employ these statistical techniques. Upon this discovery, a second round of faculty interviews were conducted. During these interviews, I asked follow-up questions to further explore why each faculty member believed the computational skill(s) in question are necessary for research in their field. If faculty's responses consisted of the statistical understandings necessary for graduate student researchers, I redirected the conversation to understand what computing skills may be required of a student to implement this type of statistical analysis with their data.

Chapter 3 reports on the data science skills outlined in phase one of this research and how they were used to tailor the existing Data Carpentry Ecology curriculum (Michonneau et al., 2019) to design workshops that suit the needs of this population of graduate student researchers. The chapter then reports on the

implementation of these workshops during the 2018-2019 academic year, describing the backgrounds and experiences of the workshop attendees. To close, the chapter outlines the next iteration of this design work, reevaluating the content of these workshops using research code produced by environmental science graduate students.

Computing Skills Employed by Environmental Science Graduate Students

With the phenomenon of acquiring the computational skills necessary for graduate-level research in the environmental sciences firmly in place (Andelman et al., 2004; Green et al., 2005; Hampton et al., 2017; Hernandez et al., 2012; Mislan et al., 2016; Teal et al., 2015; Theobold and Hancock, 2019), and an understanding of the skills environmental science faculty believe are necessary for these researchers in hand, I turned my attention to examining the data science skills employed by environmental science graduate students in their research.

Despite the elevated importance of data science to the fields of Statistics and the environmental sciences, research has yet to focus on investigating the data science skills necessary for graduate-level research in the environmental sciences. This final arm of my research focuses on using a qualitative method of investigation to describe and understand the key data science skills necessary for environmental science graduate students as they engage in the data analysis cycle.

For this research, an embedded comparative case study (Yin, 2009) was employed. This comparative case study described the key data science skills used by two environmental science graduate students, Alicia and Ellie, and compared the key skills found for each student, in the context of their educational experiences. Where the phenomenology detailed in Chapter 2 focused on describing the shared experiences of environmental science graduate students when acquiring the data science skills necessary for their research, this case study focused instead on describing the specific

data science skills used by two individuals. For this case study, Alicia and Ellie were the cases and the `R` scripts produced for their respective research were the embedded units of analysis.

At the outset of this study, a cohort of eight graduate students from environmental science fields were recruited from first semester Methods of Data Analysis courses, in the spring of 2018. These students were recruited from a variety of environmental science fields to develop an understanding of what key data science skills span across environmental science fields of research. Each of these students participated in at least two interviews, between the fall of 2018 and the fall of 2019. For the first interview, students were asked to submit all of the research code they had produced thus far. For each subsequent interview, students were requested to submit any research code they had produced since the last interview. I produced analytic memos for each of these script files, to synthesize the data science skills used throughout each student's script into higher level analytic meanings (Miles et al., 2014, p. 95). During the interview, students were then asked to describe how they learned the data science skills outlined in these memos.

When the focus came to outlining an analytical framework, however, it became clear that this initial sampling methodology aligned with grounded theory research, with the purpose of generating a substantive theory of the prevalence of specific data science skills used by environmental science graduate students in their research. Regretfully, the sampling logic of a grounded theory methodology did not align with the study's intention to intensively explore *both* the computing skills employed by students when implementing statistics in their research *and* how these skills evolve over time. Instead, an embedded case study aligns with this research goal, by selecting a few individuals and painting a picture of the data science skills they used in their research, and how each individual's skills evolved over time. Furthermore, a

comparative case study allows for the comparison of the data science skills used by each student, in the context of their personal experiences.

The rationale for selecting Ellie and Alicia were two-fold. Their experiences represent two ends of the spectrum in the computational preparation and support of environmental science graduate students as they perform data-intensive research in their field. These experiences differed in four primary ways: (1) their programming backgrounds, (2) the statistics coursework they completed for their degree, (3) the field-specific quantitative methods coursework they completed for their degree, and (4) the computing and statistical support of their adviser. Second, the research code produced by Ellie and Alicia also represents two substantially different types of computational tasks environmental science graduate students might face as they engage in the data analysis process.

Chapter 4 describes the design, analysis, and findings of this embedded case study research. Finally, Chapter 5 concludes our work and presents directions for future research outlining a learning trajectory for how students build understandings of data science concepts.

HOW ENVIRONMENTAL SCIENCE GRADUATE STUDENTS ACQUIRE STATISTICAL COMPUTING SKILLS

Contribution of Authors and Co-Authors

Author: Allison Theobold

Contributions: Designed study, collected data, performed analyses, interpreted results, and wrote manuscript.

Co-Author: Stacey Hancock

Contributions: Discussed results and implications and edited earlier manuscripts.

Manuscript Information Page

Allison Theobold & Stacey Hancock

Statistics Education Research Journal

Status of Manuscript:
_____ Prepared for submission to a peer-reviewed journal
_____ Officially submitted to a peer-reviewed journal
_____ Accepted by a peer-reviewed journal
__X__ Published in a peer-reviewed journal

Publisher: *International Association for Statistical Education & International Statistical Institute*

Submitted: January 7, 2019; Published Online: November 20, 2019; `https://iase-web.org/documents/SERJ/SERJ18(2)_Theobold.pdf?1575083627`

Theobold, A. and Hancock, S. (2019). How environmental science graduate students acquire statistical computing skills. *Statistics Education Research Journal*. `https://iase-web.org/documents/SERJ/SERJ18(2)_Theobold.pdf?1575083627`

Abstract

Modern environmental science research increasingly requires computational ability to apply statistics to environmental science problems, but graduate students in these scientific fields typically lack these integral skills. Many scientific graduate degree programs expect students to acquire these computational skills in an applied statistics course. A gap remains, however, between the computational skills required for the implementation of statistics in scientific research and those taught in statistics courses. This qualitative study examines how five environmental science graduate students at one institution experience the phenomenon of acquiring the computational skills necessary to implement statistics in their research and the factors that foster or inhibit learning. In-depth interviews revealed three themes in these students' paths towards computational knowledge acquisition: use of peer support, seeking out a singular "consultant," and learning through independent research experiences. These themes provide rich descriptions of graduate student experiences and strategies used while developing computational skills to apply statistics in their own research, thus informing how to improve instruction, both in and out of the formal classroom.

Introduction

With the increased focus on data-intensive research, statistical computing has become essential in many scientific fields. Yet, the gap between science education and students' computational knowledge has become more evident, particularly in the environmental and life sciences. The growth in computational power and the volume and variety of available data has multiplied the computational and statistical expectations of scientific researchers' abilities. Yet an abundance of literature in the environmental sciences suggests graduate students are not acquiring the

computational skills necessary for their research (Andelman et al., 2004; Green et al., 2005; Hampton et al., 2017; Hernandez et al., 2012; Lai et al., 2019; Teal et al., 2015).

Contrasted with graduate students in the biological sciences, where external structures often exist to support computational knowledge acquisition (Stefan et al., 2015), environmental science graduate students are often assumed to acquire computational skills in graduate-level statistics courses. The requirement of graduate-level statistics coursework is intended to help these students acquire the statistical knowledge necessary for their research along with any essential computational skills, but little is known about the paths graduate students actually rely upon when faced with statistical computing problems in their research. The intention of this study is to describe the experiences of graduate students in the environmental sciences to illuminate the phenomenon of acquiring the computing skills necessary to apply statistics in the context of their research. We consider the following research question: Through what paths do graduate students in the environmental sciences gain the computational knowledge necessary to implement statistics for research applications in their disciplines?

The subjects of this study were graduate students enrolled in a second semester graduate-level Applied Statistics course at a mid-size university in the Western United States. The target audience of this course is non-statistics graduate students, and, at this institution, this two-semester Applied Statistics sequence is either required or highly recommended for the completion of a master's degree in departments such as Ecology, Land Resources and Environmental Sciences (LRES), Animal and Range Sciences (ARS), and Plant Sciences & Plant Pathology. This sequence of two one-semester courses covers the foundations of statistical inference, including a wide variety of statistical methods, starting from two sample inferences and moving through regression and generalized linear models to mixed models. Taught using

an R (R Core Team, 2020) programming environment, students are typically given code to modify, covering base R graphics, data and model summaries, and built-in functions, while also being exposed to a few computational concepts such as loops, and conditional and relational statements.

The majority of graduate students in Ecology, LRES, ARS, and Plant Sciences departments enroll in the graduate-level Applied Statistics course sequence or solely in the first course in this sequence. Thus, this terminal statistics sequence often serves as graduate students' sole statistical computing course, and consequently, their only formal preparation for the computational problems they may face when implementing statistics as researchers. In examining the experiences these environmental science graduate students face when acquiring the computational skills necessary to use statistics in their research, we seek to capture an in-depth understanding of the successes and shortfalls these students encounter in their computational journey.

Though the term "Environmental Science" refers to a specific discipline in the literature, in this paper we will refer to the collection of fields that perform research in the biological and environmental sciences as "environmental science." At our institution, these are the fields whose students are required or highly-recommended to enroll in the graduate-level Applied Statistics course sequence described above. For this study, "statistical computing" is considered to consist of the computing knowledge and skills necessary for the entire process of statistical analyses, from data cleaning to data visualization to data analysis. These computing skills may include programming concepts such as loops, user-defined functions, or conditional statements, and methods for importing, cleaning, and subsetting data.

We begin by describing areas of the research literature that address the computational and statistical training of graduate students in the environmental and biological sciences. We then outline the qualitative study we implemented to

explore the experiences of graduate environmental science students in acquiring the statistical computing skills necessary for their research. The results presented reveal the prevailing experiences of these students when faced with computational problems beyond their understanding, and articulate the paths students employed to gain the required computational skills for carrying out statistics in their research.

Computing and the Environmental Sciences

Research in the computational abilities of environmental science students is in its infancy, with only a handful of institutions performing research that specifically addresses the computational training necessary to prepare students for careers post undergraduate or graduate degree. Literature related to this area has primarily focused on resources that students could potentially use to increase their computational abilities, with no studies focusing on the resources graduate students actually employ when wrestling with the computing problems necessary to apply statistics in the context of their research.

In this section, we discuss briefly three broad areas of the literature that informed this study. First, we review the literature on the foundational role computation has in the sciences. We then discuss research efforts detailing computational training in the environmental sciences, as compared with the computational training of graduate students in other biological fields. Finally, we detail research in statistics education declaring the importance of computing in the statistics curriculum.

Computing and Statistics in the Sciences

Over the last two decades, nearly every scientific field has seen a rapid increase in the use of computation and analytical tools to model phenomena across many disciplines of inquiry. In some scientific fields, such as biology and chemistry, the

recent ability to collect multitudes of data easily and quickly have made computational abilities vital to researchers and practitioners. Fields previously thought to be niche disciplines, such as computational biology, are now "becoming an integral part of the practice of biology across all fields" (Stefan et al., 2015, p. 2). Across a large sector of scientific domains, computationally heavy applications of mathematical and statistical techniques, such as management of large data sets, dynamic data visualization, and computationally intensive modeling and prediction, have become essential computational understandings for field applications (Weintrop et al., 2016). With these advances in computational power, analytical methods, and detailed computational and statistical models, scientific fields are undergoing a renaissance. These advances have, however, created a growing need for scientists to receive an appropriate education in computational methods and techniques (Fox and Ouellette, 2013; Wing, 2006).

Many chemistry, biochemistry, and bioinformatics programs have begun to incorporate computational training into their programs. A similar revolution affirming the importance of computational proficiency has yet to be experienced in environmental science fields.

Computational Training for Graduate Students in Environmental Science

The volume and variety of data collected by environmental science researchers for statistical analysis continues to increase at a rapid pace due to the availability of data from "long-term ecological research, environmental sensors, remote-sensing platforms, and genome sequencing" (Hampton et al., 2017, p. 546). These technological advances have created a crucial need to reevaluate how our system of training can better prepare current and future generations of environmental researchers (Green et al., 2005; Hampton et al., 2017).

Facing the new frontiers of "big data," programming skills to manipulate, analyze, and visualize data are becoming necessary for many ecologists. Moreover, most environmental science graduate students are required to write their own code as part of their research (Mislan et al., 2016), with the use of `R` as the "primary tool reported in data analysis increasing from 11.4% in 2008 to 58% in 2017" (Lai et al., 2019, p. 1). In a survey of a seminar course for graduate students in ecology across 11 American universities, however, Andelman and colleagues (2004) found that "ninety-three percent of students did not have skills in the scripted programming languages (e.g., SAS or MATLAB) that are needed for the integration of large datasets" (p. 244), and that one of the greatest limitations students experienced was related to data concatenation, manipulation, and analysis. Furthermore, in a recent survey of graduate students in the environmental sciences, "74% of students reported they had not completed any coursework related to the management and analysis of complex data" and only 56% of students "claimed a basic skill level in statistical applications, including `R`" (Hernandez et al., 2012, p. 1069).

This lack of computational training required for data analysis inhibits the progress of research and is laden with hidden costs. Teal and colleagues (2015) suggest that "researchers learn most of what they know about programming and data management on their own or the information is passed down within a lab" (p. 136). The costs associated with this process are substantial. Graduate students "can spend weeks or months doing things that could be done in hours or days," they may be unaware of the reliability of their results, and they are often unable to reproduce their work.

Not all biological graduate students, however, are experiencing a lack of computational training. For example, researchers in the Department of Biological and Biomedical Sciences at Harvard have developed an intensive workshop that introduces

graduate bioinformatics students to the "fundamentals of programming, statistics, and image and data analysis through the use of MATLAB" (Stefan et al., 2015, p. 2). This course is framed not only with the goals of developing programming skills and statistical understandings, but also emphasizing how to algorithmically reason through computational problems. The structure of the two-week intensive "bootcamp" consists of five full, mandatory days. The workshop dedicates the first two days to an introduction to programming using MATLAB, where students learn a variety of topics, including creating variables, performing basic variable operations, indexing, logicals, functions, conditionals, and loops. Day 3 is dedicated to developing statistical understandings, including probability distributions, hypothesis testing, p-values, bootstrapping methods, and multiple testing. Day 4 covers topics in image analysis, and Day 5 assists students in working with their own data. These workshops are given twice a year, once prior to the start of the school year as new graduate students are attending orientation, and a second time for upper-level graduate students and post-doctoral fellows (Gutlerner and Van Vactor, 2013). In introducing beginning graduate students to these concepts, researchers hoped to lower the computational barrier for students taking courses, empower students to learn computational tools on their own, and allow for other courses to "build upon this foundation and integrate quantitative methods throughout the curriculum" (Stefan et al., 2015, p. 2).

Providing effective training in data-intensive computational skills for researchers is wrought with challenges. Strasser and Hampton (2012) reported that ecology instructors indicated eight barriers to covering data-intensive computational skills. These barriers included limited time, students did not have the necessary level of quantitative or statistical skills to cover the topics, lack of resources, the instructor was not knowledgeable in these topics, topics should be included in a lab, and the

topics should be covered in other courses. These obstacles can be boiled down to "attempting to fit more material into already-full courses and curriculum, which are taught by people who do not feel prepared to address topics relevant to big data and data-intensive research" (Hampton et al., 2017, p. 547).

When considering the issue of curriculum reevaluation, however, we note that, for many environmental science fields, statistics preparation is considered vital, and statistics courses have readily been incorporated into undergraduate and graduate programs across the country.

Computing in the Statistics Curriculum

The digital age is also having an overwhelming impact on the practice of statistics and the nature of data analysis, which necessitates a "reevaluation of the training and education practices in statistics" (Nolan and Temple Lang, 2010, p. 97). The skills needed by today's statistics practitioners differ profoundly from what was needed 20 years ago. For scientific research today, computing skills are vital, especially for scientific research requiring statistical analysis (Hardin et al., 2015, p. 344).

Nearly 20 years ago, Friedman (2001) noted that "computing has been one of the most glaring omissions in the set of tools that have so far defined statistics" (p. 8). This statement is echoed in the calls from statisticians advocating for changes in the statistics curriculum (Cobb, 2015 [Discussions from Gelman, Gould, Duncan Lang, Kass, Nolan]; Nolan & Temple-Lang, 2010), as "what we teach lags decades behind what we practice" (Cobb, 2015, p. 268). Furthermore, computing has become more necessary to implementing statistical methods than even ten years ago such that "a 'just enough' level of understanding of computing is not adequate" (Nolan and Temple Lang, 2010, p. 106).

Many statisticians would agree that more computing should be included in the statistics curriculum so that students leave the classroom more computationally capable and literate. However, many statistics students are "told to learn how to program by themselves, from each other, or from their teaching assistant in a two-week 'crash course' in basic syntax at the start of a course" (Nolan and Temple Lang, 2010, p. 100). This do-it-yourself approach signals to students that statistical computing is not of intellectual importance compared to materials covered in lectures. Additionally, this structure inherits additional hidden costs, where students may pick up bad habits, misunderstandings, or the wrong concepts. Students may learn "just enough to get what they need done, but they do not learn the simple ways to do things," and the knowledge they possess when approaching a problem limits the tasks they are able to accomplish (p. 100). This brings us to question whether students in our statistics courses acquire the confidence necessary to overcome computational challenges they may face in their scientific research.

Due to the historical importance of statistics in environmental science fields, graduate students are often required or highly recommended to enroll in statistics courses for completion of their degree. As evidenced by literature in the environmental sciences, however, graduate students are not being prepared by their current curricula with the computational skills necessary to perform data-intensive environmental science research. Indeed, these commentaries by statistics educators also illuminate the lack of computational preparation with which students often leave the statistics classroom.

Methodology

In this study, we examined experiences of environmental science graduate students in gaining the computational knowledge necessary to implement statistics

in their research, and the paths that impacted these experiences. Implementation of statistics is necessary for many of these graduate students to succeed in their master's and doctoral research. Across these fields, however, students may not be acquiring these necessary skills within their graduate curriculum.

Phenomenology is a study of "people's conscious experience of their life-world" (Schram, 2003, p. 71) or their "lived experiences" (Van Manen, 1990, p. 9). As compared to case study research, which stresses the "unit of analysis, not the topic of investigation" (Merriam, 2009, p. 41, emphasis in original), a phenomenology aims to depict the essence or the structure of a shared experience through analyzing and comparing the experiences of different people (Patton, 2002).

A phenomenology was appropriate for this study, as it focuses on the experiences of graduate environmental science students as they acquire the computational skills necessary to apply statistics in their research. Participants for this study were not chosen to illustrate different aspects of a shared experience. Rather, these participants act as a cohort to illuminate and understand the phenomenon of acquiring the computational skills necessary to implement statistics through participants' lived experiences. Aspects of the backgrounds from each of the study's participants may characterize a "typical" graduate student in the environmental sciences, however, it is not the intention of these participant characterizations to focus on how backgrounds impact the experience of this phenomenon.

Participants

At our university, the two-semester graduate-level Applied Statistics course sequence (GLAS I and II) serves as a service course for graduate students in scientific fields, and only assume prerequisite knowledge of Introductory Statistics. Additionally, GLAS I serves as the required prerequisite course for other statistics

courses in the department.

Students were recruited from GLAS II in the spring of 2017. These students were interviewed following their spring break, nearly halfway through the course. Only graduate students taking the course for their respective master's or doctoral programs in environmental science fields were considered.

We requested all eight environmental science graduate students enrolled in GLAS II in the spring of 2017 to complete a survey detailing their previous statistics and computer science courses, the computer languages with which they had experiences, and their independent research experience. All eight of these students completed the survey and were then asked to participate in an in-depth interview, of which five agreed. Names of participants used in this paper are pseudonyms.

Details of the five interview participants are summarized in Table 2.1. All five identified as women, and all had taken GLAS I within the last two years. Four of the interview participants had begun or were nearly finished with their master's thesis, while Robin had just begun to work on the projects associated with her dissertation.

Of the five interview participants, Catherine's only prior statistics course had been GLAS I, Beth, Kelly, and Robin had all taken another statistics course outside of GLAS I and II, and Stephanie was completing a Graduate Certificate in Applied Statistics. The Graduate Certificate in Applied Statistics requires the completion of GLAS I and II, as well as Sampling or Experimental Design, and one additional upper-level statistics course. The Experimental Design course covers the foundations of design and analysis of experiments, including a large variety of experimental methods, starting from matrix forms and moving through factorial, balanced complete and incomplete blocking, and split plot designs. The Sampling course covers the cornerstones of sampling methodology, including a wide variety

of probability samples, from simple random sampling to systematic sampling and cluster sampling. Both courses are taught using a SAS programming environment, where students are typically given code to modify. Other courses often taken for completion of this certificate include Time Series Analysis, Multivariate Analysis, Mixed Effects Models, and Generalized Linear Models.

	Beth	**Catherine**	**Kelly**	**Robin**	**Stephanie**
Degree Seeking	MS	MS	MS	PhD	MS
Department	ARS	LRES	Ecology	LRES	LRES
GLAS I	Fall 2015	Fall 2015	Spring 2016	Fall 2015	Fall 2015
Additional Statistics Courses	Experimental Design	None	Sampling	Time Series	Time Series, Experimental Design
Languages Introduced in Coursework	`R`	`R`, SQL	`R`, SQL	`R`, SQL, Python	`R`, SQL, Python, Java
Languages Employed in Research	`R`, SQL	`R`	`R`	`R`, SQL, Python	`R`, SQL, Python
Independent Research	Thesis	Thesis	Thesis	Thesis	A few projects

Table 2.1: Academic demographics of participants: GLAS I indicates the academic semester they took the first semester graduate-level Applied Statistics course.

Over the last five years, this first semester graduate-level Applied Statistics course sequence has serviced 101 students from the departments of Ecology, LRES, ARS, and Plant Sciences. Of those 101 graduate environmental science students,

63% have gone on to complete the second semester graduate-level Applied Statistics course sequence, and only 5% have completed the Graduate Certificate in Applied Statistics.

Every interview participant from the Ecology and LRES departments voiced that they had taken a required course for their graduate coursework which introduced Access databases, providing them with experiences working with a structured query language (SQL). Robin and Stephanie continued to use SQL during their independent research and Beth learned SQL independently at the recommendation of her adviser. Unlike many environmental science graduate students, Stephanie had experience with Java from her undergraduate coursework and gained knowledge for working in Python and R from a year's work as a research assistant prior to enrolling in graduate school.

Data Collection

Following the preliminary survey, students who agreed to be interviewed were audio recorded while working through a set of ecological applications of statistical computing. These tasks assessed students' abilities to reason through applications of statistical computing, covering a broad range of problems that may be necessary for research in environmental science. These tasks were not intended to determine what statistical computing knowledge each participant did or did not possess, but rather as an entry point to capture the experiences of these participants in acquiring the statistical computing skills with which they were familiar.

After reasoning through each task, students were asked where and how they had acquired the computational skill they had employed. Based on participants' responses, the interviewer asked a follow-up question to gain additional information regarding why the participant used this resource to acquire the computational skill in question. For instance, if a participant voiced acquiring the statistical computing

skill in a course, further information was sought out regarding why she enrolled in that particular course. Alternatively, participants who voiced the Internet as their resource in acquiring the statistical computing skill were asked for additional information regarding what Internet resources they had employed. All participants were asked whether they attempted to use other resources when acquiring each skill, as well as how often they had used each resource when acquiring computational skills. Finally, every participant was asked to summarize where they have learned the computational skills necessary for implementing statistics in their research. The full interview protocol is included as an Appendix and the statistical computing tasks are included as Supplementary Materials.

The analysis in this paper is based on participant responses to questions regarding their experiences acquiring the computational skills they employed while reasoning through these statistical computing tasks.

Data Analysis

The primary author led a three-stage data analysis process (Miles et al., 2014). In the first stage, the interviews for each participant were transcribed verbatim, with participants' names removed and pseudonyms given. Subsequently, the primary author read the transcripts independently and created descriptive codes for the paths through which the participants voiced having acquired the computational skills they employed when reasoning through the statistical computing tasks. Concluding this stage, the author looked for specific references to how the courses taken by the participants had influenced their acquisition of statistical computing skills.

After working through each transcript in this manner, the primary author began the second stage of analytical coding. In this process, every path was given equal value and "nonrepetitive constituents of experiences" were linked thematically

(Moustakas, 1994, p. 96). Categories of experiences that held across multiple interviews were retained. For example, every participant voiced specific individuals they sought out as paths for knowledge acquisition. These activities were initially coded to belong to the category of "learning from others." Based on these groupings, initial categories of course work, research experience, and learning from others were constructed. Next, the primary author searched through the data to identify successes and limitations voiced by the participants when acquiring statistical computing skills within the initially identified categories. Through this step we learned that certain categories were instead subcategories, whereas others were independent of one another. For example, some participants voiced exposure to computational skills in the statistics classroom but emphasized that their understanding of these skills instead came through interactions with their peers or when using the methods in their own research. Additionally, participants who learned from others found great success in acquiring statistical computing skills from a single person in their lab or department, as compared to the limited success select participants had when using their peers to acquire statistical computing skills.

In the final analysis stage, the primary author identified emerging themes arising from these categories to describe the phenomena of acquiring statistical computing skills. The author searched for instances which reiterated the themes, as well as negative cases, with attention paid to the transcripts throughout the validation process. Following the validation process, both authors met to discuss the rationale for coding, scrutinizing the situation of each participant's description of their paths of knowledge acquisition in the context of the emergent themes. Ultimately, we reached consensus regarding the categories in which each participant's response was placed.

Although the frequency of use varied across participants, every participant voiced experiences acquiring statistical computing skills across every path, supporting

the themes that emerged. The final themes were exhaustive, mutually exclusive, and sensitizing, so that the name of the theme authentically represented the data (Merriam, 2009). These final themes present the "essence of the phenomenon" (Creswell, 2007, p. 62) of acquiring the computational skills necessary to implement statistics in environmental science fields.

Following this process, we provided the participants with the table outlining the computational skills they employed when completing the statistical computing tasks and the paths from which they voiced acquiring each skill. The participants recommended no change to be made to the table they were provided. This inclusion of member checking allows participants to check for accuracy of their statements. The ability of this study to authentically capture the experiences of students is enhanced with the lack of researcher engagement with students prior to their participation in the study. This helped to ensure that no student felt more comfortable in the interview environment, articulating their experiences, than any other student.

Results

When investigating the phenomenon of acquiring the computational knowledge necessary to implement statistics in environmental science research, we expected themes of coursework and support structure to emerge. The experiences that emerged from every participant's interview, however, related primarily to the support structures they employed, rather than the coursework that helped them to acquire the computational knowledge necessary for applying statistics in their research. In this section, we present themes describing the phenomenon of statistical computing knowledge acquisition that developed throughout the participants' interviews: (1) independent research, (2) singular consultant, and (3) peer support. A subtheme of coursework appeared within peer support and independent research,

where participants voiced the importance of their coursework on their knowledge of statistical computing. Participants consistently voiced this sub-theme to depend on either peer assistance or independent research in its impact on participants' understanding of statistical computing. The themes and sub-themes are summarized in Table 2.2.

Theme	Sub Theme	Description
Independent Research	Coursework	Research experiences that allowed students to take their course knowledge and transfer it to statistical computing applications
Singular Consultant		All-knowing past or current graduate student whom students sought out for computational assistance
Peer Support	Coursework	Assistance from peers with statistical computing tasks

Table 2.2: Participants' themes in acquisition of statistical computing knowledge.

In the sections that follow, we provide a detailed description of each theme, supplemented with quotations from participants to ensure authentic descriptions of their experiences.

Independent Research Experience

The first theme in acquiring statistical computing knowledge was participation in independent research. Involvement in independent research helped students transfer their course knowledge to statistical computing applications. This environment helped students to see the messiness of non-classroom applications and feel the unease that comes when attempting to perform statistical computing tasks beyond one's knowledge. These experiences came predominantly in the form of working as a

research assistant prior to entering graduate school, collaborating on a project in the first year of graduate school, or performing research for a master's thesis, or ultimately, a doctoral dissertation.

Catherine, a master's student in Environmental Science, who still faced everyday computational struggles, attributed the majority of her application-specific computational knowledge to her experiences in independent research. She emphasized the importance of understanding how to work in a statistical computing environment, such as `R`, which she learned by performing research, before she was able to begin to transfer the statistical knowledge she had learned in the classroom to her research:

> What I struggled with is [GLAS I] covers theory really well, but since I was new, I spent most of my time trying to figure out how to apply that theory in [`R`]. And even now I struggle transferring from R into actual statistical theory, when I'm writing my thesis. The way I had to approach it was I had to learn the `R` first, then I was able to look back on what I had actually done, in order to learn the statistics.

Kelly, an Ecology master's student, described her experiences with data management for her master's thesis as having produced the most substantial contributions to her computational abilities. Often, she attributed her intuition for solving statistical computing problems to experiences she had "merging data sets" and learning to use conditional statements for her research project. She emphasized the importance of her statistical knowledge gained in both graduate-level statistics courses in understanding "what statistical method to use," whereas she attributed becoming more fluent in statistical computing to her research experiences: "The data management stuff comes from independent research, trial and error, getting myself through." In this context, Kelly seemed to be reflecting on the computational skills she acquired when applying the statistical methods from the classroom in the context of her research, not the skills she acquired from the "trial and error"

process involved with performing research. Similar sentiments were voiced by Beth, an Animal & Range Science master's student, who attributed nearly all of her computational knowledge as having stemmed from her independent research. With the recommendation of her adviser, she taught herself how to create an Access database to store her data. In storing her project data in this manner, she was able to learn important concepts about data structures, subsetting data "using qualifiers and criteria," and sorting data, skills which were then easily transferred into `R` to manage data for analysis.

Singular Consultant

When describing who they seek out for computational help, every participant described first seeking out an "all-knowing" past or current graduate student. These individuals served as "singular consultants," with whom these students had the "best," most productive experiences in finding solutions to computational problems that had arisen in their implementation of statistics to their research. For Beth, this singular consultant came in the form of a past graduate student from Animal & Range Sciences who was hired to help faculty complete projects:

> We have a guy who used to be a student in our department and then he was hired on again to help finish some projects, after he got his master's in Statistics. He is very helpful with [pointing out what's wrong with your code]. He's very good with code and if I have a quick question he can always answer it.

For Kelly, another graduate student on her same project served as this consultant. Kelly described turning to this particular graduate student for help with computational problems she had encountered in her thesis; she added that other graduate students in their department also used this person as a consultant for their computational problems:

> The other grad student on this project is so well versed in `R` that he's unofficially become the person that people go to with questions.

Throughout her computational struggles, Catherine found assistance from previous graduate students from the department, but she found the most assistance from a previous graduate student "who had left the department and was off professionally somewhere else, but he still took the time to help walk me through [my code]."

One participant, Stephanie, an Environmental Science master's student, served as this singular computational consultant for many members of the Environmental Science department. With her experiences teaching herself `R`, she was able to "explain code in a way that makes sense," says Robin, a fellow Environmental Science doctoral student who has often sought out help from Stephanie. With an adviser from a computational background and a project which required sophisticated statistical modeling, Stephanie "had to learn to code." Additionally, her laboratory often worked in collaboration with Computer Science faculty, where she and her lab-mates were taught computer science coding practices and jargon. "Stephanie has gotten good at teaching it, because everyone on our floor is like 'I can't do this, Stephanie help me'," said Robin. Stephanie stated that graduate students have sought her assistance "daily" or "at minimum two to three times a week." In contrast, when Stephanie experiences difficulty in performing computational tasks, she has found solace in her lab-mates and ultimately, when necessary, with her adviser:

> My entire lab works in the same room and my adviser's door is always open. So, if someone is having a major issue, whoever is in the room can hear that. If [my adviser] hears me ask [a lab-mate] how to do something and he knows how, he just shouts how to do it. So, it's a very group oriented dynamic. I've never had to go beyond the people in my lab.

Peer Support

The third theme in acquiring computational knowledge that all participants spoke of was the support they had received from fellow graduate students when performing computational tasks related to applications of statistics. The students described how, when they are unsure of how to complete a computational task for their research and their singular consultant is not available to them, they turn to fellow graduate students for help. Participants described instances when the computational tasks required of them were beyond their current knowledge or occasions when they had been unsuccessful at attempting to complete a problem and sought out help from a fellow graduate student. For example, Kelly, an Animal & Range Science master's student, shared that when she reached a point in coding when she didn't know how to do something, she turned to one of her lab-mates:

> I've been to a point where I didn't know how to do something with my knowledge or what I can find online, and then I'll go to one of my lab-mates.

Catherine, a master's student in Environmental Science, spoke of the expectations of her advisers that the computational problems she was being asked to perform were "easy, since she had all the information." Catherine has had numerous experiences, however, where she did not have the knowledge necessary to perform the task or she was missing "little caveats" that kept her from fully being able to perform the tasks. When faced with these problems, she "reached out to previous students that had taken the course."

Robin, a doctoral student in Environmental Science, reiterated Catherine's experiences, describing how she reached out to other graduate students in other labs for help with computational problems. Alternatively, Stephanie, as a singular consultant, voiced that when she was faced with computational problems beyond her

knowledge, she had never been forced to "go beyond talking to her lab-mates" for assistance.

Unfortunately, peer support did not always provide an optimal solution. This may be a potential reason that participants sought help from peers only when their singular consultant was unavailable. For example, Kelly described negative experiences when seeking computational assistance from graduate students not of close proximity to her:

> When I'm struggling with something and I go to other grad students, they'll say "I did this the other day. I'll send you my code." I've found most of the time I don't understand what they've done enough to plug in what I want and make it work. There have been a few times when making tables and plots and someone sends me their code and I can just plug in my data and it works just fine. I've had less success with that.

Discussion

The present study, although exploratory in nature, outlines the experiences of environmental science graduate students to shed light on the phenomenon of obtaining the computational skills necessary to apply statistics in the context of environmental science research. The themes identified, and their corresponding examples, illustrate the essence of the structure of the shared experience of these participants. These results help to illuminate the gaps that exist between the statistical computing skills these students acquire through their curriculum and the computing skills required for them to successfully implement statistics in their research.

Our expectation of coursework to be a primary source of statistical computing knowledge was not found for these participants. When these graduate students encountered a statistical computing problem, they would pull upon the knowledge they had acquired through their graduate coursework, but this knowledge was often insufficient. Rather, the computational understandings that these students

attributed to their statistics coursework were primarily low-level concepts, such as using built-in `R` functions, adding comments to their code, and limited trouble-shooting of error messages. Additionally, these low-level concepts were said to only be fully understood through participants' peer interactions, or as they were being implemented independently within their own research.

Instead, participants voiced that having experiences performing independent research substantially influenced their abilities to reason through and perform the computational tasks required for various statistical analyses. Through independent research, the participants were able to play with real-world data and applications more complex than what they had encountered in the classroom. The programming skills developed during a student's independent research, in conjunction with peer collaboration, were described largely as high-level concepts, such as conditional statements, loop implementation, and user-defined functions. Students described their independent research as having opened the door to experiencing the unease that comes when one is asked to perform statistical computing tasks beyond one's knowledge, a feeling they had not encountered in their courses. In these circumstances, students stated that they would ask for help from the people with whom they had the most prior success or felt the most comfortable.

In a direct connection to the participants' discomfort in asking for help from an adviser, the theme of a singular consultant emerged. These singular consultants served as an "all-knowing" individual, from whom the participants had either had the "best" experiences with, where the individual spent the necessary time to explain the concepts, or the consultant had always been capable of providing the participant with a solution to their problem. These individuals served as the first line of defense when statistical computing problems arose, where participants were both able to seek computational help and acquire new computational skills and understandings through

their interactions. If this consultant was unavailable to the graduate student due to time or physical constraints, these students then turned to their peers.

Peer support was initially discussed by the participants in their interviews as a mechanism they used when their "code doesn't run" or when they were asked (or needed) to do something beyond their current computational understandings. However, this theme continued to emerge as the participants worked through computational problems, often attributing their knowledge of a computational procedure to a friend or fellow graduate student helping them "do it with their data." These peers offered a path for students to seek help, often voiced to be more comfortable than asking an adviser, where participants described both the fear of asking and "feeling dumb" or being "brushed off" because their adviser thought they should "be able to figure out how to do it." As opposed to the help participants received from their singular consultant, these students also voiced negative experiences they had encountered when seeking help from their peers, such as a peer sending them "helpful" code that they did not understand.

Lastly, the adviser played an important role in students acquiring the computational knowledge necessary to perform applications. Despite students' reluctance to seek out computational assistance from their adviser, advisers did often emphasize the importance of statistical computing skills, as well as introduce (or recommend) students to store their data using a relational database. The participants' ability to understand both data structures and sorting or filtering data was largely attributed to their experiences working with these types of databases. Although these interviews found that advisers were often considered as the last line of defense, they were, however, viewed as an accessible way for students to better understand the statistical computing necessary for their independent research projects, which overall contributed to better computational understanding and skills for these students.

Implications

The implications for statistics and environmental science education focus on identifying and understanding the importance of the computational knowledge necessary to apply statistical methods in environmental science research, and the paths graduate students employ to acquire these essential skills. Environmental science fields have long understood the importance of statistics education for their students, so a preponderance of programs recommend or require at least one graduate-level statistics course. Conversely, many of these graduate programs are not actively incorporating computational courses into their degree, instead assuming that students are acquiring these skills in their recommended statistics courses. Unfortunately, computational skills necessary for research are not typically included in these statistics courses (Friedman, 2001; Hardin et al., 2015; Nolan and Temple Lang, 2010). As evidenced in the research on computational preparation of environmental science students (Andelman et al., 2004; Green et al., 2005; Hampton et al., 2017; Hernandez et al., 2012; Mislan et al., 2016; Teal et al., 2015), the experience of poor computational preparation is not unique to students at this institution. A restructuring dilemma is faced by both fields—statistics education and the environmental sciences—with intractable differences between the curricula of statistics service courses and the expectations of environmental science research.

Implications for Statistics Educators

Statistics educators should consider the power an applied statistics course sequence has to provide graduate students with a year-long introduction to statistical computing. As seen by Stephanie, who entered graduate school after completing a year's work as a research assistant working in `R`, these learning experiences can

help to alleviate the power differential students feel when asking their advisers or peers for assistance. However, the content covered by graduate applied statistics sequences is expected to paint a vast picture of the field of Statistics, with topics ranging from a difference in means to mixed-models. Consequently, many educators feel they do not have the time to incorporate statistical computing into the classroom, and some feel that they have limited computational expertise to teach these concepts (Hampton et al., 2017; Nolan and Temple Lang, 2010). The inflexibility of graduate programs further complicates this issue, as many graduate students are unable to enroll directly in a statistical computing course due to an already full and demanding course load. Thus, questions should be raised about how to best bridge this gap between coursework and research expectations for statistical computing skills.

The importance of playing with statistical applications on real-world data, as voiced by these participants, should also be considered by statistics educators at all levels. This transition to incorporating authentic, research-like tasks, which engage students in statistical computing, can be supported by online resources, data-discovery tools, example datasets and code, and instructional tools, along with collaborative course designs and the sharing of instructional materials.

Implications for Environmental Science Educators

Due to the extensive research on computational preparation of environmental science graduate students, faculty in these fields have a growing awareness of these issues of computational ill preparation. Yet, most of this research has focused on a vast array of computational skills students do not possess, rather than focusing on the computational skills necessary to implement statistics in their research. Environmental science faculty should thus have an increased awareness of the statistical computing preparation with which graduate students leave the statistics

classroom. As echoed by the participants in this study, the implementation of statistics in the context of environmental science research is not always as tidy as is presented in the classroom. Hence, to better support these students' acquisition of the computational skills necessary for implementing statistics in their research, additional preparation focusing specifically on statistical computing should be considered by faculty in these fields.

The impact of an undergraduate education on students' experiences as graduate researchers should be considered by all statistics and environmental science faculty in higher education when recognizing the importance of developing data-intensive statistical computing skills early on in undergraduate statistics courses. In this study, none of the participants voiced having any experience working with `R` in their undergraduate coursework. Instead, these students encountered `R` for the first time in their first semester of graduate school during the first graduate-level Applied Statistics course. The participants who had computing experiences in their undergraduate coursework or post baccalaureate research work or experience with Access databases were able to navigate learning `R` with greater ease than students with no computing experiences. This lack of computing experience was further compounded when students began their independent research, where students with fewer computational skills and understandings had substantially different independent research experiences than their counterparts with more. The frustrations of simple tasks, such as subsetting data or removing NA's, were felt by the participants who had completed a bachelor's without any computational elements to their coursework, whereas those who were exposed to even small amounts of computing in their undergraduate coursework were able to begin computational tasks in their research walking and not crawling.

Limitations and Future Research

Although the methodology we used to describe the phenomenon of acquiring the computing knowledge necessary to implement statistics for graduate students in the environmental sciences provided important themes of knowledge acquisition, it is not without its limitations. Eliciting descriptions of computational knowledge acquisition yielded varied experiences with each of the main themes, but richer data could be gathered in a future longitudinal study. Following graduate students throughout their program of study could further identify where students are acquiring statistical computing knowledge, as well as instructional methods that best assist students in obtaining these understandings. To better inform environmental science and statistics faculty, a thorough investigation of both the coursework and structure of courses completed by these participants could be performed. This would allow for a discussion of how to best integrate these computational concepts into current coursework requirements, so that students leave the classroom with understandings they can implement immediately in their own research.

The focus of this study of environmental science graduate students' experiences acquiring the statistical computing skills necessary for their research should not be generalized to experiences acquiring general computing or programming skills. Whereas general programming skills may overlap with statistical computing skills, the foundation of study of each set of skills differs. Rather than focusing on computer architecture, design, and application, statistical computing skills center around the study of data. Select universities have, however, begun to require general computing courses for undergraduates majoring in environmental science fields (Cortina, 2007; Rubinstein and Chor, 2014; Wilson et al., 2008). The doors to future research will open as these students begin to enroll in graduate programs in environmental sciences. This future research can instead focus on understanding how students transfer their

general programming knowledge to acquiring statistical computing knowledge, and which skills possess the greatest overlap.

Conclusion

Statistical computing has become a foundational aspect of research in the environmental sciences. This small-scale exploratory study brings forward the experiences of graduate environmental science students in acquiring the computational understandings necessary to successfully perform statistical applications for independent research. Participants found the greatest success in acquiring the computational skills required for their research through independent research, a singular consultant, and peers. Whereas others have noted the importance of integrating computing into the statistics curriculum (Friedman, 2001; Hardin et al., 2015; Nolan and Temple Lang, 2010) or the lack of computational preparation for environmental science graduate students (Andelman et al., 2004; Green et al., 2005; Hampton et al., 2017; Hernandez et al., 2012; Mislan et al., 2016; Teal et al., 2015), we instead explored the phenomenon of acquiring the computational knowledge necessary to implement statistics in graduate environmental science research. The computational burdens experienced by these participants when implementing statistics in the context of their research and the computational understanding with which they left the statistics classroom suggest the need for integration of formal computational training into these programs. The present study helps to emphasize the importance of computing skills necessary for data-intensive environmental science research.

Acknowledgements

We would like to specially thank the participants from this study, without whom this research would not have been possible. We would also like to thank Mary Alice Carlson, Jennifer Green, Megan Higgs, Megan Wickstrom, co-editor Jennifer Kaplan, assistant editor Beth Chance, and reviewers for their insightful comments on this paper.

DESIGNING DATA SCIENCE WORKSHOPS FOR DATA-INTENSIVE ENVIRONMENTAL SCIENCE RESEARCH

Contribution of Authors and Co-Authors

Author: Allison Theobold

Contributions: Designed the study, taught the workshops, collected the data, performed the analysis, interpreted the results, and wrote the manuscript.

Co-Author: Stacey Hancock

Contributions: Discussed the workshop design, the methods of data collection, and edited earlier manuscripts.

Co-Author: Sara Mannheimer

Contributions: Collaborated in applying for National Network of Libraries of Medicine grant for external funding for workshops, and edited final manuscript.

Manuscript Information Page

Allison Theobold, Stacey Hancock, & Sara Mannheimer

Journal of Statistics Education

Status of Manuscript:
_____ Prepared for submission to a peer-reviewed journal
__X__ Officially submitted to a peer-reviewed journal
_____ Accepted by a peer-reviewed journal
_____ Published in a peer-reviewed journal

Publisher: Taylor & Francis Group, LLC

Submitted: December 15, 2019

Abstract

Over the last 20 years, statistics preparation has become vital for a broad range of scientific fields, and statistics coursework has been readily incorporated into undergraduate and graduate programs. However, a gap remains between the computational skills taught in statistics service courses and those required for the use of statistics in scientific research. Ten years after the publication of "Computing in the Statistics Curriculum," the nature of statistics continues to change, and computing skills are more necessary than ever for modern scientific researchers. In this paper, we describe research on the design and implementation of a suite of data science workshops for environmental science graduate students, providing students with the skills necessary to retrieve, view, wrangle, visualize, and analyze their data using reproducible tools. These workshops help to bridge the gap between the computing skills necessary for scientific research and the computing skills students leave their statistics service courses with. Open to faculty, staff, and the larger community, these workshops promote continued learning of the tools necessary for working with data and provide additional resources for incorporating data science into the classroom.

Introduction

Scientific fields have seen profound increases in the volume and variety of data available for analysis. Matched with the growth in computational power, today's scientific researchers are faced with computational and statistical expectations beyond those of the coursework dictated by their curriculum. In the environmental sciences, though statistics courses have been readily incorporated into undergraduate and graduate curricula, an abundance of literature suggests that these curricula fail to equip graduate students with the computing skills necessary for research in their field

(Andelman et al., 2004; Green et al., 2005; Hampton et al., 2017; Hernandez et al., 2012, Mislan, Heer, & White, 2016; Teal et al., 2015; Theobold and Hancock, 2019). Only one of these studies (Theobold and Hancock, 2019), however, acknowledges the substantial role statistics courses could potentially play in students' acquisition of computational skills.

Over the last 10 years, a large number of statistics educators have echoed Nolan and Temple Lang's call to "embrace computing and integrate it fully into statistics undergraduate major and graduate programs" (Nolan and Temple Lang, 2010, p. 97; Baumer, 2015; Baumer, Horton, & Wickham, 2015; Cetinkaya-Rundel and Rundel, 2018; Cobb, 2015; Hardin et al., 2015; Horton and Hardin, 2015; Kaplan, 2018; McNamara and Horton, 2018). Indeed, the American Statistical Association Curriculum Guidelines for Undergraduate Programs in Statistical Science (2014) reflect the increasing importance of data science skills. Despite this campaign for computing in the statistics classroom, graduate-level statistics service courses have largely been overlooked, even though their potential impact is substantial. Unlike courses designed for an undergraduate or graduate program in Statistics, these service courses often act as the sole exposure to computing with data prior to the start of a student's independent research.

The intention of this research is to (1) describe the computing skills necessary for research in the environmental sciences, (2) investigate how these skills can be infused into currently existing extracurricular workshops, and (3) understand the experiences of attendees of these workshops. We consider the following research questions:

1. What do environmental science faculty members identify as the key data science skills necessary for graduate students to engage in the entire data analysis cycle?

2. How can currently existing workshop materials be tailored to meet the needs of environmental science graduate students?

3. What are the backgrounds and experiences of individuals attending data science workshops as a means to acquire data science knowledge and skills?

We investigated these areas of interest using a three-phase design-based implementation research model (Fishman et al., 2013). In the first phase, we conducted in-depth interviews with faculty from environmental science fields regarding the computational skills they believe are necessary for graduate students to succeed in their research. Phase two then focused on adapting currently existing workshop resources to design of a series of data science workshops targeting the key computational skills distilled from these interviews. The final phase consisted of implementing the workshops and collecting survey responses from the workshop attendees regarding their experiences participating in each workshop.

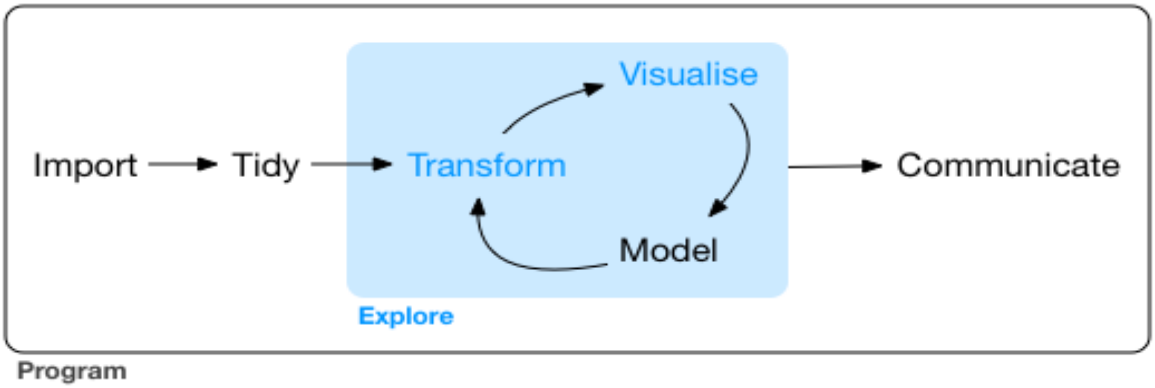


Figure 3.1: Data Analysis Cycle, Wickham, H. & Grolemund, G. (2017) *R for Data Science*. Sebastopol, California: O'Reilly.

For this research, the collection of disciplines who perform research across a variety of environmental science fields are captured under the term "environmental science." At our institution, these are the fields of Ecology, Land Resources and

Environmental Sciences, Plant Sciences and Plant Pathology, and Animal and Range Sciences, whose students are required or highly recommended to complete graduate-level statistics coursework for a masters or doctoral degree. In this paper, the "data analysis cycle" consists of all stages in the data analysis process, from data importation to data exploration to the communication of results (Figure 4.1), where data modeling is but one component. The "data science skills" necessary to engage in this cycle may include general programming concepts such as loops, user-defined functions, or conditional statements. However, the cornerstone of data science skills differs fundamentally from general programming skills, with a focus on data rather than computer architecture, design, and applications.

We begin by outlining areas of research that address the computational and statistical training of graduate students in the environmental sciences and the potential for extracurricular workshops to fill gaps in students' computational preparation. Next, we outline methodology used to design and implement a suite of data science workshops tailored to environmental science graduate students. Next, we summarize the first phase of research, outlining the computational skills faculty members identified as necessary for graduate students to succeed in their independent research. We then outline how these identified skills were interwoven into existing data science workshop materials for researchers in the environmental sciences. The next section summarizes the backgrounds and experiences of the workshop attendees during the 2018-2019 academic year, and describes the research conducted on the implementation of the workshops. We reflect on the resources that have facilitated the sustainability of these workshops, alongside possibilities for formally integrating these workshops into the university curricula. With the first iteration of this research complete, next we outline future research plans for a second iteration of this design based implementation research. To close, we revisit the current climate of computing

in the statistics curriculum for service courses and describe how these types of extracurricular workshops can assist in further integration of computing into these classrooms.

The Current Climate of Statistics and Computing in the Environmental Sciences

Due to the substantial growth in the volume and variety of available data over the last two decades, the practice of environmental science has changed dramatically. Advances in technology have made computationally heavy applications of data science techniques—such as management and coalition of large data sets, high frequency spatial and temporal data visualization, and hierarchical Bayesian modeling—essential understandings for environmental science research. This flood of data has "challenged the research community's capacity to readily learn and implement the concepts, techniques, and tools" (Hampton et al., 2017, p. 546) necessary for data-intensive environmental science research, creating a crucial need to re-evaluate how our educational system can better prepare current and future generations of researchers (Green et al., 2005; Hampton et al., 2017).

Computing in the Environmental Sciences Curriculum

Arising from a decade of mumblings about the importance of computing to research in the environmental sciences (Andelman et al., 2004; Dodds et al., 2007, 2008; Eglen, 2009; Green et al., 2005; Hastings et al., 2005; Kelling et al., 2009; Wilson, 2006; Wilson et al., 2008; Wing, 2006), 2010 brought two studies on the computational ill-preparation of environmental students by their curriculum. In the first large scale study of ecology instructors, Strasser and Hampton found that undergraduate students were not being prepared with the data management tools necessary to engage in environmental science research. Across 51 different institutions, despite largely

affirming the importance of data management skills, fewer than 20% of instructors reported data management topics in their courses. That same year, an environmental science graduate student led a large scale study of the computational experiences of future environmental scientists (Hernandez et al., 2012, p. 1068). In a survey of environmental science graduate students across the United States, the authors found that over 74% of the students surveyed reported they had no skills in any programming language—including `R`—and only 17% reported basic skill levels in any programming language. Hence, a large number of students may be leaving their undergraduate and graduate programs without the data science skills necessary for data-intensive research in their field. Hernandez and colleagues, however, noted that student-focused workshops could work toward bridging this gap, by "providing intensive environments" where students could learn "particular methods or technologies" (p. 1075).

Due to the lack of "training in data and computing skills" (Data Carpentry, 2020, p. 136) in undergraduate and graduate programs in the environmental sciences, external learning opportunities are necessary to prevent researchers from continuing to teach themselves or each other everything they know about data management and analysis. Out of these need for high-quality resources for learning scientific computing emerged The Carpentries project (2019). The Carpentries focuses on teaching "foundational computational and data science skills to researchers" through in-person, hands-on, domain-specific workshops. As part of their educational mission, The Carpentries collaboratively develops publicly available lessons for populations of researchers, which do not assume that attendees have any prior knowledge before attending the workshops. Teal and colleagues acknowledge that, while these workshops "will not be able to teach researchers all of the skills they need in two days," workshops "are a way to get started," lowering the activation energy required

to begin acquiring computing skills and empowering researchers "to be able to conduct the analyses necessary for their work in an effective and reproducible way" (p. 143).

Over the last 20 years, statistical preparation in the environmental sciences has grown to be considered vital (Hampton et al., 2017), and statistics coursework has been integrated into graduate programs across the nation. Yet, none of these conversations have acknowledged the substantial role students' statistics education potentially plays in their attainment of the data science skills necessary for research. Today, throughout their research, the majority of environmental science graduate students are required to produce code as part of their data analysis process (Mislan et al., 2016). To compound the difficulties these graduate students face, over this same period, the software used by environmental science researchers has shifted, with an increase of over 45% in the use of `R` in environmental science publications since 2008 (Lai et al., 2019). The clear need for data science proficiency in environmental science research necessitates a transformation of the environmental science curriculum similar to that which infused statistics preparation into the required graduate coursework.

Computing in the Statistics Curriculum

Changes in the digital age have also had "a profound impact on statistics and the nature of data analysis" (Nolan and Temple Lang, 2010, p. 97), with today's skills differing substantially from what was needed but five to ten years ago. In the year following the publication of "Computing in the Statistics Curriculum" (Nolan and Temple Lang, 2010), the Mckinsey Report (Manyika et al., 2011) was published. The McKinsey report stated that, by 2018, "the United States alone could face a shortage of 140,000 to 190,000 people with deep analytical skills as well as 1.5 million managers and analysts with the know-how to use the analysis of big data to make effective decisions" (p. 3). With calls to transform the undergraduate statistics curriculum

resounding nationally, the 2014 American Statistical Association (ASA) President, Nathaniel Schenker, convened a workgroup to update the association's guidelines for undergraduate programs. These new guidelines included an increased emphasis on data science skills and real applications, specifically students' ability to "access and manipulate data in various ways, use a variety of computational approaches to extract meaning from data, program in higher-level languages" (American Statistical Association Undergraduate Guidelines Workgroup, 2014, p. 7).

With this curricular momentum, in 2015, *The American Statistician* produced a special issue on "Statistics and the Undergraduate Curriculum," to encourage submissions of broader topics in the statistics curriculum. Articles in the special issue ranged from detailing how computing should be included throughout the Statistics curriculum (Green and Blankenship, 2015; Tintle et al., 2015; Hesterberg, 2015), to presenting thoughts on how data science topics should be integrated into undergraduate statistics courses, (Nolan and Temple Lang, 2015; Grimshaw, 2015; Baumer, 2015; Hardin et al., 2015). In the same issue, George Cobb provocatively stated that the statistics curriculum needed to be rebuilt "from the ground up" (2015), as "what we teach lags decades behind what we practice" and "the gap between our half-century-old curriculum and our contemporary statistical practice continues to widen" (p. 268). Moreover, despite the issue's focus on the broader statistics curriculum, statistics educators continued to lament that the current Introductory Statistics curriculum teaches a snapshot of the entire data analysis cycle, "wherein challenges with data computational methods, and visualization and presentation are typically elided" (Baumer, 2015, p. 336).

The following year, however, brought the revised GAISE college report (Revision Committee, 2014), creating a push for reform in the Introductory Statistics curriculum. The six recommendations originally outlined by the committee in 2005

continued, but the authors suggested two new emphases for the first recommendation (teach statistical thinking), which better reflect the modern practice of statistics. First, statistics educators should "teach statistics as an investigative process of problem-solving and decision making," and should "give students experience with multivariable thinking" (2014, p. 3). These recommendations reiterate the sentiments heard throughout the statistics community, that students should emerge from our courses with the understanding that data analysis "isn't just inference and modeling, it's also data importing, cleaning, preparation, exploration, and visualization" (Cetinkaya-Rundel, 2018). Yet, the inclusion of these topics in the Introductory Statistics curriculum is still a heated discussion. Many educators continue to believe (1) that it is not possible to teach statistical concepts and programming in just one course, (2) that teaching programming takes up valuable time which could be used towards teaching important statistical concepts, or (3) students are not interested in learning to program (Cetinkaya-Rundel, 2018). Thus, despite charges for the statistics community to "treat computing as fundamental as basic mathematics and writing" (Nolan and Temple Lang, 2015, p. 298), many students leave their Introductory Statistics course without "a set of practices and attitudes about data that are immediately applicable to their lives" (Gould, 2010, p. 309).

Amidst these conversations, `R` packages were being created, which would fundamentally changing how users interact with `R`. These `R` packages, universally known as the "`tidyverse`," have created user friendly `R` tools which "share an underlying design philosophy, grammar, and data structures" (Wickham, 2017). Statistics educators have begun to leverage these tools in the Introductory Statistics classroom to teach reproducibility (Baumer et al., 2014), data management (Baumer et al., 2015), dynamic data (Hardin, 2018), and big data (Wang et al., 2017). While there exists a growing momentum to incorporate these new `R` tools into the

Introductory Statistics classroom, attention has yet to be paid to alternative statistics service courses, such as those taken by environmental science graduate students. These courses, like Introductory Statistics, serve graduate students from a variety of scientific backgrounds. However, unlike an undergraduate Introductory Statistics course, students are expected to emerge from their statistics coursework with the statistical and data science skills necessary for their research.

The frustrations echoed by environmental science educators (Hampton et al., 2017; Teal et al., 2015) suggest that, despite the inclusion of statistics coursework into these graduate programs, students continue to leave the statistics classroom without the data science skills necessary to participate in the data analysis cycle in their own research. The fundamental question raised ten years ago by Nolan and Temple Lang still applies today: do our students leave the statistics classroom able to "compute confidently, reliably, and efficiently?" (2010, p. 100). An in-depth study of environmental science graduate students' experiences acquiring the computing knowledge necessary for their research answered this question with a resounding 'no' (Theobold and Hancock, 2019). Like the hypothesis of Teal and colleagues (2015), these students did not attribute their acquisition of the data science skills necessary for their research to the statistics courses they took for their degree. Rather, students gained the data science skills necessary to engage in the entire data analysis cycle through independent research experiences, an "all-knowing" past or current graduate students, and peer networks. Ten years after the publication of "Computing in the Statistics Curriculum," we continue to assume that "students will 'pick up' the skills they need" to participate in the data analysis cycle outside of their statistics coursework (Gould, 2010, p. 309).

Extracurricular Workshops to Bridge the Gap

Reiterated by both statistics education and environmental science researchers alike (Nolan and Temple Lang, 2010; Teal et al., 2015), this lack of training in computational skills impedes the progress of scientific research, sends the signal to students that computing is not of intellectual importance, and is laden with hidden costs. Students may pick up bad habits, misunderstandings, or the wrong concepts, learn just enough to get what they need done, spend weeks or months on tasks that could be done in hours or days, and they may be unaware of the reliability and reproducibility—or lack there of—of their results (Nolan and Temple Lang, 2010, p. 100; Teal et al., 2015, p. 136). But why are these skills still so rarely included in these service courses when the need for them is widely recognized?

Environmental science educators have reiterated the challenges in integrating computing into the curriculum outlined by Nolan and Temple Lang. These barriers can be boiled down to "attempting to fit more material into already-full courses and curriculum, which are taught by people who do not feel prepared to address topics relevant to big data and data-intensive research" (Hampton et al., 2017, p. 547). These hurdles are potentially even greater for graduate-level statistics service courses. Instructors of these courses are often explicitly told the statistical content students are expected to learn, while it is implicitly assumed they are also teaching students the data science skills necessary for them to participate in the entire data analysis cycle. Claiming these graduate students ought to take additional, data science specific courses to obtain these skills is infeasible for many, as graduate programs frequently leave little room for additional coursework.

Until computing has been meaningfully integrated into these service courses, extracurricular workshops hold the potential to address the gap between the computational preparation of students by their coursework and the computational

requirements of their research. Although data science skills can potentially be acquired from the drove of currently available online resources, such as online lessons, MOOCs and books, none of these resources provide researchers with the ability to pose their questions directly to an instructor or to learn from others. Moreover, this drove of online materials, poses a "significant challenge in being able to discover relevant and high-quality materials," for researchers with limited time.

As, reiterated by Nolan and Temple Lang (2015), extracurricular learning opportunities are not a direct substitute for the prolonged instruction of these skills that occurs in a course. However, this is not the goal of these learning opportunities. Instead, short, intensive workshops, such as those provided by The Carpentries, are able to teach immediately useful skills that can be taught and learned quickly, keep learners active by using live coding and formative assessment, work with a learners from a variety of backgrounds, and build learners' self efficacy (Word et al., 2017), so that attendees "learn the computational aspects as part of an interesting, challenging, and confidence-building process" (Nolan and Temple Lang, 2010, p. 101).

Methodology

Improving environmental science graduate students' access to "powerful, effective learning opportunities" (Fishman et al., 2013, p. 137) necessitates understanding the skills required for these students to be successful in their research. Design-based implementation research (DBIR) (Cobb et al., 2003; Fishman et al., 2013; O'Neill, 2012) "offers a model for the design and testing of innovations within the crucible of classrooms and other contexts for learning" (Fishman et al., 2013, p. 140). Rather than creating workshops covering content outside parties believe are important, DBIR uses collaboration with members of the community to develop "evidence-based improvements" (p. 143) to teaching innovations—situating community members as

"co-designers of solutions to problems" (Fishman et al., 2013, p. 140) rather than bystanders. Furthermore, DBIR emphasizes the iterative process of design and evaluation, and thus is particularly well suited for research that develops "evidence-based improvements to innovations," where evidence from the implementation informs changes made to these innovations for learning.

This paper describes the results of the first iteration of this DBIR, consisting of three phases. Table 3.1 outlines the phases of this research, the research question each phase addresses, and the data collected during each phase. First, we summarize computing skills necessary for graduate-level environmental science research which were investigated during phase one. Phase two of this research details how the skills identified during the first phase were used to tailor currently existing Data Carpentry (2020) and Software Carpentry (2020) lessons to meet the needs of environmental science graduate students. Finally, we chronicle the third phase of this research, implementing and evaluating these workshops. This final evaluation phase focuses on the survey results of the backgrounds and experiences of workshop attendees, rather than the workshop content or learning outcomes of attendees, which are described as directions for future research.

Theories of Learning

Engaging in design based implementation research necessitates a thorough outlining of the theories of learning which stand as the foundation from which the teaching innovation is built. The design of these workshops was informed by three primary channels: (1) prior research on graduate students pathways for computational knowledge acquisition, (2) workshop materials collaboratively created by environmental science researchers and educators, and (3) educational research on the importance of hands-on learning environments.

In interviews with environmental science graduate students, it became clear that students were not acquiring the computing skills necessary for their research directly through

Research Question	Design Phase	Data Collected
What do environmental science faculty members identify as the key computing skills graduate student require to implement statistics in environmental science research?	Phase 1	Faculty interviews
How can the key computing skills identified by environmental science faculty be incorporated into currently existing workshop materials?	Phase 2	Carpentries curriculum materials
What are the experiences of individuals attending data science workshops?	Phase 3	Pre- and post-workshop surveys

Table 3.1: Research questions, phases, and data collected for the three-phase DBIR.

their coursework (Theobold and Hancock, 2019). Instead, students reported either teaching themselves through their research experiences or learning through their peer networks. Two of these participants stated that they had attempted to gain computing skills by attending an in-person workshop, but had little to no success because the workshop was not tailored to their learning needs. Kelly, an Ecology master's student stated "I went to one `R` workshop, last semester and it was way over my head. I got a little out of it. I was able to follow along, but it was too much for my level." Additionally, many participants remarked on the difficulties they had experienced when attempting to Google solutions to the computational tasks they were facing. Beth, an Animal and Range Sciences master's student lamented "I think Googling stuff really can make it confusing. A lot of times when I Google it I end

up on some crazy StackExchange thred. I read the answers on the blog post and I have no idea what they're talking about."

Both of these women's experiences reiterate concerns raised by environmental science educators (Teal et al., 2015) regarding the difficulties researchers face when attempting to acquire computational skills. While there are a large number of online resources for learning to data science skills in `R`, "there is a significant challenge in being able to discover relevant and high-quality materials and for already busy researchers to commit their time and focus to these learning activities" (Teal et al., 2015, p. 136). In-person workshops address these concerns by providing students with a "short, focused time" to "work on developing new skill sets" (2015, p. 137), while explicitly outlining the skills that will be taught in each workshop. Moreover, workshops that are designed for individuals with little to no prior computing experience allow for learners to "self-select whether or not they should attend," and set a "clear expectation for the pace of instruction" (2015, p. 138). Lastly, in-person workshops promote deeper learning, with participant's ability to ask questions to an instructor or assistants and learn from others'.

The content of these in-person workshops was informed by the curriculum developed by Data Carpentry (2020) and Software Carpentry (2020). Statisticians and environmental scientists (National Academies of Sciences, Engineering, and Medicine, 2018; Teal et al., 2015) agree that data science instruction ought to be grounded in contextual examples, because "people learn best when new skills are building on an existing framework" (Teal et al., 2015, p. 138). Data Carpentry workshops are tailored to be domain-specific, so "researchers can learn more quickly and effectively," and they can "see more immediately how to implement these skills and approaches in their own work" (2015, p. 136). Originating from Software Carpentry, the Data Carpentry Ecology curriculum has been developed by the ecology community, to "share perspectives on best practices" (2015, p. 137), and is taught across the world, so "there are opportunities for feedback and refinement of the lessons to deliver a higher quality product" (2015, p. 137). The Data Carpentry Ecology

curriculum represents a set of publicly vetted materials that teach the key data science skills necessary for environmental science research, as these materials are open source and developed collaboratively with the community. This "vetting" allows anyone to propose an improvement, which is then reviewed, improved, and merged into the core curriculum, "so that everyone can benefit from better explanations, examples, and exercises" (2015, p. 137). Hence, the open source materials available through the Data Carpentry Ecology curriculum represent the "best" theory for the teaching and learning of data science skills this field possesses.

The importance of active learning across learning settings has resonated across the discipline of education. Thus, congruent with the nature of the materials developed by The Carpentries, these workshops make extensive use of hands-on learning opportunities. Hands-on learning "gives researchers the chance to develop their computational skills in the course of the workshop, so they leave with practical examples and hands on experience" (Teal et al., 2015, p. 137). Learning to program is often frustrating, but "when embedded in exploring data, drawing plots, looking for anomalies, making conjectures, and looking for supporting evidence," learners build their computational skills "as part of an interesting, challenging, exciting, confidence-building process" (Nolan and Temple Lang, 2010, p. 101). For many novice programmers, "it is a big leap from practicing basic programming skills to embracing problem-solving methodologies and general computing principles" (Nolan and Temple Lang, 2010, p. 101). However, by building experiences where learners "behave like scientists who work with data" (p. 101) and exposing students to the creativity involved in computing with data, this leap is lessened.

Computing Skills Necessary for Environmental Science Research

As the direct supervisors of graduate students, environmental science faculty members are potentially aware of the computing skills that are vital to researchers in their respective fields. Thus, interviews with faculty members from these fields allow for us to

gain an understanding of the essential skills required of environmental science graduate students.

In the spring of 2017 and fall of 2018, every faculty member in the Ecology, Land Resources and Environmental Sciences, Animal and Range Sciences, and Plant Sciences and Plant Pathology departments, currently overseeing a graduate student were emailed requesting their participation in this research. While some faculty enthusiastically agreed to participate, others declined for three main reasons—they hadn't directly overseen a graduate student recently, they deemed themselves to be weak in statistics, or they were unavailable to meet. Table 3.2 outlines the number of faculty requested for participation and the number of faculty interviewed, by departmental affiliation.

Department	Faculty Invited	Faculty Interviewed
Animal & Range Sciences	7	2
Ecology	15	8
Land Resources and Environmental Sciences	24	8
Plant Sciences & Plant Pathology	15	5

Table 3.2: Number of faculty members requested for participation and interviewed, by department.

Data Collection

Faculty agreeing to participate were interviewed regarding (1) the computational skills they believe are necessary for masters and doctoral students to implement statistics for research in their field, and (2) how they believe graduate students acquire these necessary skills. The full interview protocol is included in Appendix A.

Based on faculty's responses, the interviewer asked follow-up questions to further explore why the faculty believe the computational skill(s) in question are necessary. For instance, if a faculty voiced the need for students to be able to build a data workflow, further information was sought regarding what specific computing skills this would require. Alternatively, when the response from faculty consisted of the statistical understandings

necessary for graduate student researchers, follow-up questions were asked to delve further into what computing skills a student may require to successfully implement this type of statistical analysis with their data. Not only did these interviews provide valuable feedback on *what* content the workshops should include, they also added insight into *why* workshops form the ideal mode of delivery for this needed training.

Data Analysis

The primary author led a three-stage data analysis process (Miles, Huberman, Saladaña, 1994). During the first stage, the interviews for every faculty member were transcribed verbatim. Following this process, the primary author read the transcripts independently, highlighting excerpts where computing skills were discussed. The author then created descriptive codes for the skills faculty identified as necessary in each of these excerpts. At the close of this stage, the author examined these codes for specific references to computing skills currently addressed in Data Carpentry's *Data Analysis and Visualization in `R` for Ecologists* lesson. This lesson "uses a tabular ecology dataset from the Portal Project Teaching Database and teaches data cleaning, management, analysis, and visualization" (Michonneau et al., 2019).

Following this process, the primary author began the second stage of analytical coding. This stage acts as a method of synthesizing descriptive summaries, tying together "different pieces of data into a recognizable cluster," demonstrating how the data are instances of a general concept (Miles and Huberman, 1994, p. 95). During this stage, skills were linked thematically, and themes that held across multiple interviews were retained. For example, every faculty voiced students' need to work with data in `R`. This theme was initially labeled "working with data," with additional themes of data wrangling, and data visualization created. Next, the author searched through these themes to uncover how each theme related to the others. Through this process it was determined that certain themes captured similar constructs, and were merged into a single theme, whereas other constructs were voiced independently, and separate themes were formed. For example, while every

faculty voiced students' need to work with data in R, these sentiments were voiced alongside students' need to perform other data wrangling operations, such as reorganize data, filtering out rows of data, selecting columns, creating new variables, or modifying existing variables. Hence, the themes of "working with data" and "data wrangling" were merged into the single theme of "working with data." Alternatively, while reproducibility is a key aspect to working with data in R, the skills identified by faculty that this theme captures were not voiced alongside a specific software. Instead, when these faculty commented on the need for students' work to be reproducible, R was continually mentioned as the vehicle to support this need.

In the final stage of the analysis, the primary author searched the faculty transcripts for evidence supporting the emerging themes, scrutinizing whether each identified skill fit into the existing themes. Following this validation process, the first and second authors met to discuss the rationale for each code and inspect the skills identified by faculty in the context of the emergent themes. These final themes ground the theory for creating an effective intervention promoting the acquisition of computing skills necessary for graduate-level environmental science research.

Skills Identified by Environmental Science Faculty

While some faculty had difficulties disentangling the statistical methods students use in research from the computing required to implement those methods, many were able to express the computing skills necessary for graduate students in their field to engage in the entire data analysis process. A substantial overlap was seen between faculty expectations and the "data acumen" outlined by NAS (National Academies of Sciences, Engineering, and Medicine, 2018), falling into three categories: (1) working with and wrangling data, (2) data visualization, and (3) reproducibilty.

Working with Data Every faculty member interviewed believed that students' experiences in the statistics classroom do not adequately prepare students to work with

and organize large, messy datasets. As graduate students perform their research, they are required to think about "storing data, managing data, matching data, and collating data," into a meaningful datasets for analysis. Some faculty, aware of the different types of data their students work with, reflected that it is "not uncommon to be analyzing half a million records, but I think it's uncommon to be doing it effectively or efficiently."

These skills for working with data ranged from students' ability to "organize their data and get it in a way that can be used by `R`" to tasks that required reorganizing data formats from wide to long or vice versa—a skill which every faculty member griped is not acquired through the standard curriculum. A faculty member bemoaned that standard examples in statistics courses provide students with data which are the product of cross-tabulation, so students are never forced "to figure out how to get the cross-tabulation [they] need, so that [they] can bring it into `R` and do [their] regression." These concerns reiterate the importance of "data management and curation" detailed by NAS, who stated that "at the heart of data science is the storage, preparation, and accessing of data" (National Academies of Sciences, Engineering, and Medicine, 2018, p. 26).

Data Visualization The importance data visualization plays in every stage of students' research was emphasized by every faculty member interviewed. Faculty affirmed that students should possess the ability to create visualizations of their data early and often. These expectations align with the the facility outlined by NAS (2018), who stated that students need to have the ability to "present data in a clear and compelling fashion" (p. 26). One faculty member declared that students' ability to look at their data in different ways dramatically shapes their research potential, and the tools available today allow for researchers to create visualizations precisely tailored for each investigation. Many faculty voiced the usefulness of the `ggplot2` package (Wickham, 2016) lowering the barriers for students to learn "how to visualize [their] data to explore and understand it."

Reproducibility Every faculty member emphasized the usefulness of "manipulating data in ways that are repeatable," through scripted programs such as `R`. Across environmental science disciplines, faculty concurred that many students do not use `R` for data wrangling, and instead rely on Excel because "the code (`R`) is kind of a black box" and when they "don't have that instant connection with [their] data, I think it fundamentally boils down to fear." Concerns were raised for the students using unreproducible tools to wrangle their data, as "they would never find [their] way back to what the original data set would have been" and their advisers would have no way to understand why certain data are missing. While many advisers stated that they encourage students to avoid these brute force Excel manipulations, they reflected that students may not have the computing skills necessary to perform the same data wrangling task in a scripted and reproducible manner. These faculty concerns parallel the "workflow and reproducibility" acumen outlined by NAS, who stated that students need to "be exposed to the concept of workflows" (National Academies of Sciences, Engineering, and Medicine, 2018, p. 28).

How Students Gain Computational Skills Across environmental science disciplines, faculty stated that they assume students are acquiring the computing skills necessary to analyze their data either in their required statistics coursework or on their own. When asked why students are not acquiring computing skills in their field-specific courses, a faculty member stated, "We don't really have anyone to teach that. It's not that it isn't valuable, but there is no one to teach it." Some faculty believed "most graduate students come in knowing more about the tools one might use to manipulate data than their advisers do," while others lamented the gaps between the computational skills of their graduate students and their own training, feeling "personally out of touch [with students] because I haven't taken the time to learn `R`, because of my training and my age."These gaps impact the assistance faculty can provide to their students, as "increasingly faculty feel that they're not at the forefront of their programming abilities, so their students are being self-taught and are often computationally ahead of them."

Designing Data Science Workshops for Graduate Students in the Environmental Sciences

The second phase of this research attended to the development of a suite of data science workshops targeted to graduate students in the environmental sciences. Skills identified through faculty interviews were incorporated into a set of four 3-hour workshops covering (1) the basics of programming in `R`, (2) intermediate programming tasks in `R`, (3) creating appropriate and effective data visualizations, and (4) cleaning and merging data in preparation for analysis and visualization, all using reproducible tools.

The materials for these workshops were adapted from Data Carpentry's *Data Analysis and Visualization in `R` for Ecologists* lesson (Michonneau et al., 2019), a curriculum maintained by experienced researchers in ecological fields, which uses ecology-specific data contexts [1]. Importantly, the first workshop in *Data Analysis and Visualization in `R` for Ecologists* does not assume that attendees have any previous experience working in `R`, and each workshop builds on knowledge acquired at previous workshop(s), without the expectation that attendees have acquired additional knowledge or skills between workshops. The workshop materials developed for this research are available through GitHub [2], with video tutorials recorded and available through our institution's library [3].

Data Context

Emphasized by both the NAS and these faculty, "effective application of data science to a domain requires knowledge of that domain" (National Academies of Sciences, Engineering, and Medicine, 2018, p. 29). Hence, data science instruction ought to be grounded in "substantive contextual examples," to "ensure that data scientists develop the capacity to pose and answer questions" with data relevant to them (2018, p. 30). Therefore,

[1] This work is a derivative of *Data Analysis and Visualization in `R` for Ecologists* (https://datacarpentry.org/R-ecology-lesson/) by Data Carpentry, used under CC-BY (https://creativecommons.org/licenses/by/4.0/).

[2] GitHub repository (link)

[3] MSU Library videos (link)

ecological data are used for these workshops, originating from Montana Fish, Wildlife, and Parks and the Portal Project Teaching Database (Ernest et al., 2018). These data highlight a variety of aspects that commonly occur in ecological data, including multiple sampling instances, mark-recapture, biological measurements, and meta- and micro-level data.

The data from Montana Fish, Wildlife, and Parks contain entries of 18,352 fish caught over the span of ten years on the Blackfoot River in central Montana. Each row of the data contains information for a single captured fish, and the columns represent: the trip number; section of river sampled; length, weight, and species of the fish; and whether the fish had been previously captured. The Portal Project Teaching Database contains three separate data files of time-series data collected on a small mammal community in southern Arizona: micro-level data on the rodents captured on individual plots; macro-level data on rodent species and taxa; and macro-level data on the treatments applied to different plots.

Computing Tools for Environmental Science Research

The structure and context of these workshops include a statistical programming language used extensively throughout environmental science research (`R`), environments which facilitate the learning of `R` (RStudio and RStudio Cloud), and tools that promote reproducibility throughout the entire data analysis cycle (R Markdown).

Why `R`? The use of `R` is widespread throughout the environmental science research community, a dramatic change over the last decade (Lai et al., 2019). Furthermore, with the invention of the RStudio IDE (RStudio Team, 2015b), this user-ship continues to increase, as `R` includes over 100 packages frequently used in ecological data analysis (https://CRAN.R-project.org/view=Environmetrics). `R` is free and open source, so attendees learn a statistical programming language that will be accessible to them throughout their careers. Unlike MARK, VORTEX, or RMAS, with `R`, attendees results do not depend on remembering the sequence of buttons they clicked. With the shocking realization that large numbers of modern scientific findings cannot be replicated (The Economist Editorial, 2013; Johnson,

2014) and the growing appreciation for reproducible data analysis methods in ecological research (Cassey and Blackburn, 2006; Ellison, 2010; Morrison et al., 2016; Powers and Hampton, 2019), today's researchers in scientific fields are becoming more aware of the need for a reproducible data analysis workflow.

Why RStudio? RStudio is a free computer application that allows you access to the resources of `R`, while providing you with a comfortable working environment (RStudio Team, 2015b). The RStudio IDE "makes [programming] less intimidating than the bare `R` shell" (Cetinkaya-Rundel and Rundel, 2018, p. 59). Additionally, the RStudio environment is consistent across operating systems, which is not the case for other statistical software packages. Moreover, because RStudio is an IDE, it includes integrated help files, intelligent code completion, and syntax highlighting—all of which help to reduce the learning curve.

Why RStudio Cloud? The RStudio Cloud was created as a platform to make it easy to do, share, teach, and learn data science using `R` (RStudio Team, 2015a). Through the Cloud, attendees are able to access publicly available workshop materials, without worrying about software installation, package installation, or transferring data. Workshop participants interact with the workshop's materials in the same manner as a locally installed version of RStudio, as seen in Figure 3.2, and an organized RStudio project directory exposes attendees to best practices for reproducible project construction.

Why R Markdown Documents? R Markdown documents provide an easy-to-understand framework to combine statistical computing and written analysis in a single document, helping to break the copy-paste paradigm for generating statistical reports (Baumer et al., 2014). During the workshop, R Markdown documents allow for attendees to keep their code organized and their workspace clean, which is unnatural for new learners. Each workshop's master R Markdown document contains blocks of code and descriptions for every topic covered, allowing for participants' exploratory work to be saved within a topic. For additional information on R Markdown documents see Baumer et al. (2014).

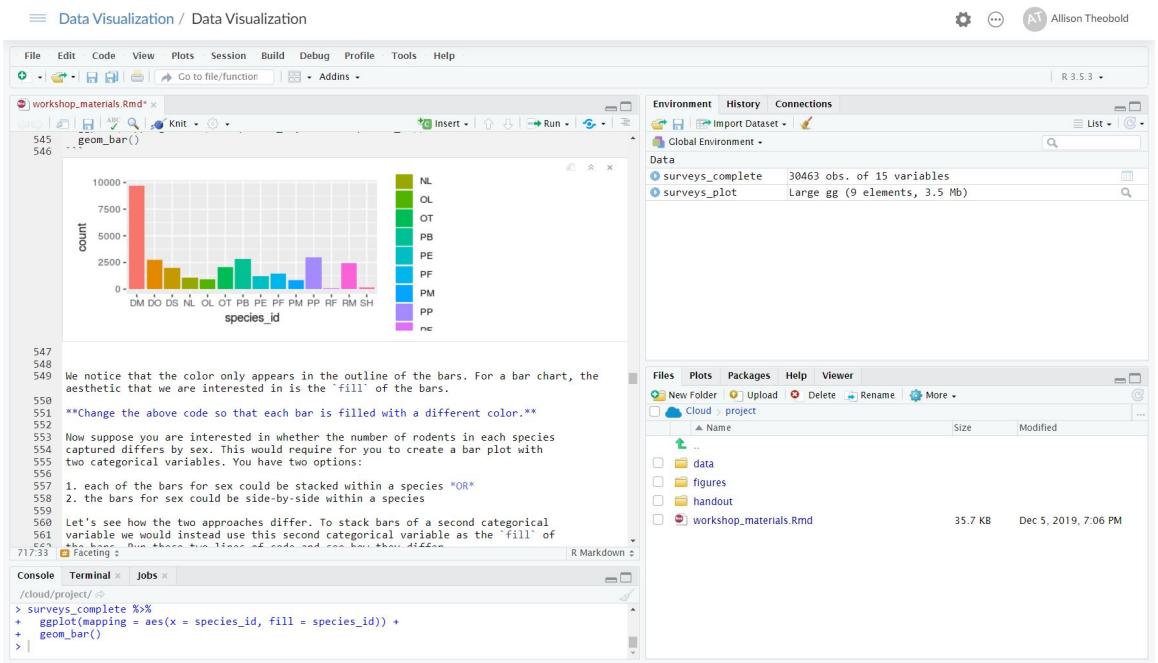


Figure 3.2: RStudio Cloud workspace environment for *Data Visualization with `ggplot2`* workshop. Every workshop works in an RStudio project, containing a master R Markdown file, a data folder containing the data used in the workshop, and the handout produced for attendees.

Workshop Content

For the creation of these workshops, we used the current materials for learning Ravailable through Data and Software Carpentry. As mentioned previously, Data Carpentry has developed an ecology specific curriculum, with a lesson specifically for learning to work in R. The *Data Analysis and Visualization in `R` for Ecologists* lesson is comprised of six sub-lessons: Before We Start, Intro R, Starting with Data, Manipulating Data, Visualizing Data, and R and SQL. All of these sub-lessons except R and SQL were used to create this workshop series. For skills which faculty identified that were not taught in the *Data Analysis and Visualization in `R` for Ecologists* lesson, we used materials from Software Carpentry's *`R`for Reproducible Scientific Analysis* lesson. This lesson provides a more general introduction to working in R, with 16 different topics. Included in these topics are sub-lessons targeting software skills, namely Subsetting Data, Control Flow, and Functions Explained.

Table 3.3 outlines the sub-lessons included in each of the tailored workshops, and how the original Data Carpentry of Software Carpentry content was modified in the production of the workshop.

Workshop	Carpentries Lesson(s) Used	Additions	Deletions
Introduction to `R`	(DC) Before We Start, Intro `R`, Starting with Data	`list` data types	subsetting vectors, formatting dates
Intermediate `R`	(SC) Subsetting Data, Control Flow, Functions Explained	logicals	subsetting lists, nested for-loops, while-loop, defensive programming, using `source()` to load functions
Data Wrangling	(DC) Manipulating Data	formatting dates, working with characters, data joins	
Data Visualization	(DC) Visualizing Data	histograms, density plots, barcharts	

Table 3.3: Lessons from Data Carpentry and Software Carpentry used in the creation of these workshops, DC represents sub-lessons from the Data Carpentry *Data Analysis and Visualization in `R` for Ecologists* lesson and SC represents sub-lessons from the *`R` for Reproducible Scientific Analysis* lesson.

Introduction to `R` This first workshop in the series covers the basics of learning to program in `R`. The workshop first introduces the RStudio environment and project work flow in RStudio, discussing project working directories and relative paths. Next, the workshop progresses through tools for working with vectors and lists of different data types, motivating methods for working with dataframes. After learning how to import data

into R, the workshop proceeds through inspecting data, extracting data, and changing data types. Motivated by working with missing data, the workshop introduces R help files to inspect function arguments and their default values. These help files are called upon as participants make use of base R functions to create data summaries, perform simple data cleaning, and produce both univariate and bivariate visualizations of the data.

Intermediate R This second workshop covers coding skills to modularize R code. The content in this workshop, excluding relational statements, is not included in Data Carpentry's *Data Analysis and Visualization in R for Ecologists* lesson. Instead, many of these concepts are covered in Software Carpentry's *R for Reproducible Scientific Analysis* lesson. Yet, conditional statements, for-loops, and user-defined functions are skills that many environmental science faculty asserted were necessary for graduate students to possess as they perform independent research.

First, the workshop then progresses through the use of relational statements and how to link these statements using and (`&`), or (`|`), and not (`!`) conjunctions. Next, the workshop dives into the use of conditional statements, stepping from `if`, to `if else`, to `else if` statements. The second half of the workshop covers methods to iterate or replicate a set of instructions many times. Looping, specifically `for()` loops, are introduced as a popular way to iterate or replicate the same set of instructions. Working through exercises which repeat operations on a dataset using both a `for()` loop and a recursive `for()` loop, motivate a discussion of why R users recommend the use of vectorization for non-recursive `for()` loops. To conclude, functions are presented as an approach to replicate the same set of instructions in multiple locations throughout your code. Motivated by a script which copies and pastes the same process multiple times, attendees understand why this is an undesirable practice. Attendees are then tasked with transforming the copy-paste-modify process into a function. By parsing out the function writing process into a set of steps that should be used when you have copied and pasted your code multiple times, participants leave with a foundational understanding of why functions are useful and practical approaches for

implementing them in their own code.

Data Wrangling with `dplyr` and `tidyr` The *Data Wrangling* workshop introduces common data wrangling issues faced by environmental science researchers. Inspired by the difficulty of reading bracket subsetting and how cumbersome it can be to remember the different base `R` functions and formats to wrangle your data, this workshop introduces the `dplyr` (Wickham et al., 2018) and `tidyr` (Wickham, 2014) packages. Much of `R`'s language has not changed over the last 20 years, which leaves the desire for a "smoother, more efficient, and more readable pipeline for modern `R` workflows" (Ross, Wickham, & Robinson, 2017, p. 19). The `tidyverse` packages share common interfaces and data structures that make it simpler to learn data wrangling tasks and allow for the process to flow naturally from one step to the next.

The workshop begins by outlining six of the common "verbs" that handle common data wrangling challenges, included in the `dplyr` package: `select()`, `filter()`, `mutate()`, `group_by()`, `summarise()`, and `arrange()`. Prompted by the need to perform a sequence of multiple data wrangling operations, participants learn how to connect each of these data wrangling verbs using the pipe operator (`%>%`). Next, motivated by the need to integrate additional data files for analysis, the concept of relational data is outlined. After an introduction to key-value pairs, attendees make use of the `left_join()` and `right_join()` functions to join these additional data files.

The final topic of the workshop involves the issue of data reorganization. Until now, participants have been presented with "tidy" data, where every observation is one row, each variable has a column, and every value has one cell. This concept of "tidy" data is used to describe 'long' and 'wide' data formats. The `tidyr` package is introduced to alleviate the burden of data reorganizations, when transforming data from one layout to another. In groups, participants work through a final exercise summarizing groups, using `pivot_wider()` to spread these values across multiple columns, and finally using `pivot_longer()` to gather these multiple columns into a single column.

Data Visualization with `ggplot2` The final workshop in the series dives into creating data visualizations using the `ggplot2` package (Wickham, 2016). Rather than remembering a list of functions that make different visualizations, each with its own unique syntax, arguments, inputs, and outputs, `ggplot2` creates a uniform interface with functions that each solve a particular class of problems. This uniform syntax and "vocabulary for describing the elements of a statistical plot" (Nolan and Perrett, 2016, p. 261), allows participants to create more dynamic visualizations out of the gate.

Using the joined data from the close of the *Data Wrangling* workshop, a scatterplot is used to illuminate a discussion of the `ggplot()` syntax. Participants learn about the `mapping` argument for specifying aesthetics (`aes`) for the plot and the set of `geom` functions which define the type of plot you produce. By making explicit connections between the addition operator (`+`) and the pipe operator, participants understand addition to be an intuitive metaphor for adding layers to a plot. Next, the workshop examines how to modify the `ggplot()` aesthetics and geoms to create violin plots, density plots, bar charts, and line plots, allowing for participants to explore the `geom` functions and aesthetics that pair with each plot. A conversation is had about the importance of plotting raw data rather than simply aggregate measures of the data, and the difficulties that might arise. Similar to the advise of Nolan and Perrett (2016), adding a `geom_point()` or `geom_jitter()` layer to a visualization highlights tools that can be used so graph elements don't interfere with the data (e.g. jittering, transparency). Finally, faceting, using `facet_wrap()` and `facet_grid()`, is introduced as an additional visualization tool to facilitate multivariate comparisons (Nolan and Perrett, 2016, p. 261).

By this point in the workshop, participants have posed many questions on how to modify aspects of a plot that don't depend on the geom. For the final section of the workshop, the group walks through different customizations one can make to each `ggplot` object, to add clarity and information to the plot. Participants learn how to flip a plot's coordinate, how to make customizations of the plot's labels, the size of the points, the

thickness of lines, the appearance of the plotting window, and the color scheme used. Each of these customizations continue to emphasize to participants the iterative nature of creating data visualizations, transforming a simple plot step-by-step "into a graph that is data rich and presents a clear vision of the important features of the data" (Nolan and Perrett, 2016, p. 262).

Evaluating Data Science Workshops

During the 2018-2019 academic year, a total of 202 students, faculty, and staff attended at least one of the workshops. During the fall and spring semesters, a total of 84 individuals attended the *Introduction to* `R` workshop, 74 attended *Intermediate* `R`, 20 attended *Data Wrangling*, and 24 attended the *Data Visualization* workshop. The *Introduction to* `R` and *Intermediate* `R` workshops were offered twice during the fall semester, and once during the spring semester. The *Data Wrangling* and *Data Visualization* workshops were only offered once during the spring semester. The first workshop was offered two weeks after the start of the semester, with three week breaks taken between each of the subsequent workshops. Each workshop lasted a total of three hours and was taught by one lead instructor with two to three workshop assistants.

Data Collection

In the week prior to the workshop, a survey is sent out to those registered for the workshop through a public Google Form. The pre-workshop survey details individuals' areas of study, current occupations, statistics and computer science experiences, participation in independent research, and their chosen method of storing any data they've collected. Following each workshop, attendees are asked to complete another survey. This post-workshop survey details the workshop participants' experiences in the workshop environment, their familiarity with the content covered in the workshop, their perceived ability to implement the skills they learned, the best thing about the workshop, and what

could use improvement. Of the 202 workshop attendees during the 2018-2019 academic year, we obtained 121 complete pre-workshop surveys and 56 complete post-workshop surveys.

The content of these surveys was informed from the pre- and post-workshop surveys disseminated for Data and Software Carpentry workshops [4], with revisions to the disciplines and occupations provided, and removal of questions regarding the degree of agreement with statements provided. The full pre- and post-workshop surveys are included in Appendix D and Appendix E While attendees rate their perceived ability to implement the concepts learned in the workshop to their own research, these surveys cannot speak to the learning outcomes of workshop attendees. Furthermore, aside from participants perceived ability to implement the workshop material, the surveys do not provide information regarding whether each attendee acquired the data science skills they were in search of. Due to the importance of addressing these issues when developing and implementing external workshops, these interests are outlined as directions for future research.

Data Analysis

At the close of each semester, attendees' answers to pre- and post-survey free response questions were qualitatively analyzed. For each of these questions, surveys were inspected and initial descriptive codes were created for each response. For the statistics experiences, descriptive codes were produced based on the the name of the course, the content of the course, and the reported course number and institution. Descriptive codes for what attendees attested they hoped to learn, what they enjoyed, and what they reported could be improved focused on the content and context of the response.

In the final production of analytical codes for each of these four questions, themes were created to be mutually exclusive and exhaustive, where the name of the theme authentically represented the attendees' description (Merriam, 2009). For example, one

[4]This work is a derivative of the Carpentries pre- and post-workshop survey materials (https://github.com/carpentries/assessment/), used under CC BY (https://creativecommons.org/licenses/by/4.0/).

attendee stated they were hoping to "improve my abilities to effectively use R," which was assigned to the theme of "interest in learning more about R." A participant reporting a "very welcoming atmosphere and fun group of people" was categorized as belonging to the theme of "workshop atmosphere." Another attendee stated that "more time was needed to cover the workshop material adequately. And really, I personally do not mind a 4 hour session with a tea break in between," which was categorized to belong to the theme of "time."

Backgrounds of Workshop Participants

The majority of the workshop attendees were from environmental science fields—from departments such as Land Resources and Environmental Sciences (LRES), Ecology, Plant Sciences and Plant Pathology, Biochemistry or Microbiology, Animal and Range Sciences, and Earth Sciences. Additionally, over 60% of workshop attendees were masters and doctoral students. It is worth noting, however, that 18 faculty, staff, and postdocs also attended these workshops. Figure 3.3 displays the department affiliations of the workshop attendees and their current occupation.

Consistent with the environmental science literature (Andelman et al., 2004; Hampton et al., 2017; Hernandez et al., 2012; Teal et al., 2015), a large number of workshop participants were either unfamiliar with the concept of a programming language or had no experience with any programming languages. Nearly 60% of attendees reported no experiences with any programming languages, 20% reported experiences working in R, and 30% reported experiences with other programming languages (e.g. MatLab, SQL, Java, C).

Many attendees, however, stated that they had taken courses in statistics. The majority of participants reported having either undergraduate or graduate experiences with an introductory level statistics course. Notably, over 15% of attendees reported having no formal statistical training. Most graduate students had enrolled in discipline-specific introductory statistics courses in their own department or a graduate-level applied statistics course offered by the Department of Mathematical Sciences. Table 3.5 consolidates themes of workshop participants' previous statistical experiences, when asked to report the statistics

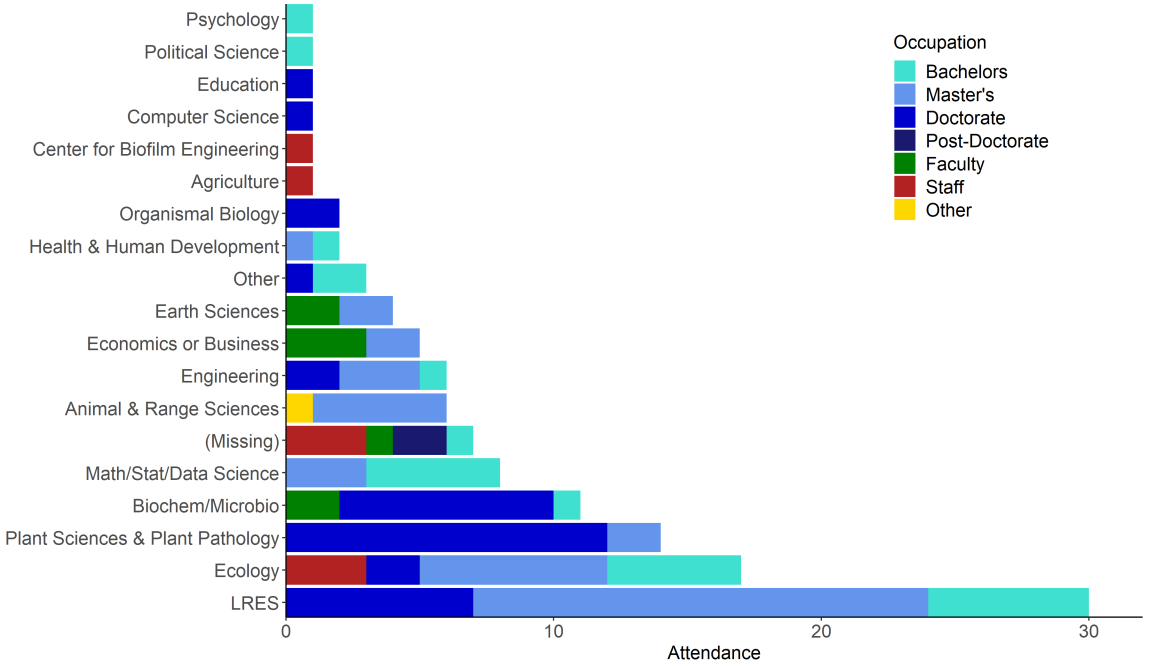


Figure 3.3: Number of attendees by department and current occupation, selected from an itemized list of campus departments and positions.

Programming Languages	Participants
What is a programming language?	30
None	35
R	22
SQL	12
Java or Javascript	11
C or C++	7
Fortran	4

Table 3.4: Workshop attendees' responses to the question of "What programming languages do you have experience with? Select all that apply."

courses they have taken over the course of their education.

Motivation for Attending

As expected from the prevalence of the use of R in environmental science research (Lai et al., 2019; Mislan et al., 2016), over half of the master's, doctoral, and post-doc

Stat Courses	Participants
Introductory Statistics	46
Applied Statistics	42
None	24
Discipline Specific Introductory Statistics	20
Intermediate Statistics	10
Experimental Design	8
Probability Theory	6
Statistical Computing	3
Sampling	3
Biostatistics	2
Spatial Analysis	2
Econometrics	1
Time Series Analysis	1

Table 3.5: Workshop attendees' responses to the question "What are your previous statistical experience(s)? List course names.," thematically organized based on content of the course.

workshop participants attended for assistance with their research. Others were seeking additional assistance for learning the `R` skills necessary for their coursework, refreshing or updating their `R` skills to include new tools they were unfamiliar with(e.g. `ggplot`, `dplyr`), or undergraduates preparing for graduate school.

Reason Attended	Participants
Research assistance	58
Coursework assistance	35
Refresh or update skills	16
Department/Professor recommended	13
Preparation for graduate school	12
Professional Development	7
Adviser recommended	6
Expand Skills	6

Table 3.6: Workshop attendees' responses to the question "What is your most important reason for attending this workshop? Select all that apply."

As echoed by previous studies of environmental science graduate students (Teal

et al., 2015; Theobold and Hancock, 2019), attendees overwhelmingly stated that they primarily use the internet (27%), their peers (21%), or their lab mates (15%) when learning R. Based on the statistical backgrounds of these participants and the statistics education literature on computing in the statistics classroom, it is not surprising that nearly two-thirds of these individuals reported using resources other than course materials as their main resource for learning R.

Resources Used to Learn R	Participants
Internet Resources	55
Peers	43
Course Materials	35
Lab Mates	29
Adviser	20
These Workshops	15
Books	3
Professor	1

Table 3.7: Workshop attendees' responses to the question "What resources have you used while learning to program in R? Select all that apply."

Reflections of Workshop Participants

The percentage of individuals reporting that all of the information presented was new to them differed by workshop, with 40% of *Introduction to R* participants, 30% of *Intermediate R* participants, 80% of *Data Wrangling* participants, and 50% of *Data Visualization* participants stating the information was new to them. Across every workshop, nearly all participants stated that they "strongly agreed" that they "learned skills that [they] will be able to use in [their] research/work." Over 75% of the workshop participants reported that they would use the skills they learned in their research immediately or in the next 30 days.

The themes which emerged from these attendees' reflections to what they enjoyed most about the workshop were hands-on learning, workshop atmosphere, instructor

attributes, and confidence. Many attendees felt walking through the code step-by-step and the hands-on exercises "fostering a much greater level of understanding" and left them feeling more "confident figuring things out on my own." Furthermore, these attendees voiced that the workshop left them feeling more independent, because "I have a better understanding of how to read code, what certain symbols/terms/etc mean and how they work." Individuals who reported using the internet as a resource to learn `R` stated that "it's easy to walk away from `R` workshops wondering if anything was learned, however the exercises were a clear tool which allow me to see what I gained."

Changes to Future Workshops

While the feedback from workshop attendees reinforced many of the current qualities of the workshops, some comments provoked directions for future changes. These changes fell into four main categories: timing, resources, interactivity, and content. When asked what aspect(s) of the workshop could use improvement, many attendees suggested revisions to the duration of the workshop. Many individuals stated that "it could be better to have more time," suggesting an additional hour with a break in between. However, others reflected that it could be more beneficial to have "shorter but more workshops," since "it's easy for the brain to get tired after an hour." These comments led us to one immediate change and one future change. During the 2019-2020 academic year, each workshop continued to last a total of three hours, however, we added additional breaks during the workshop. Following the exercise associated with each topic, approximately every 15-20 minutes, the workshop took a brief break, where attendees could stand up, stretch, and ask questions. Additionally, when changing instructors, approximately every 45 minutes, the workshop took a 5-10 minute break.

A future change, due to the attendees' feedback, is to transition the beginning "housekeeping" explanations to a 5-10 minute pre-workshop video. Although the majority of workshop attendees have little to no programming experiences, few if any questions arise during the first 5-10 minutes of the workshop, which lead us to consider encapsulating

this content into a video. In these first minutes, attendees are led through the layout of RStudio, given an explanation of the structure and purpose of an R Markdown document, and shown how to execute `R` code, with a demonstration of what happens when lines of code are executed. For the *Introduction to `R`* workshop, each of these tasks could be rolled-up into a 5-10 minute video with an interactive R Markdown file, for attendees to watch and interact with before they attend the workshop. Eliminating this tutorial from the beginning of the workshop clears up additional time that can be spent covering workshop content.

Due to over 60% of workshop attendees reporting they use the internet or their peer networks when learning to program in `R`, changes were made to the handout that attendees receive at each workshop. Each handout for the 2019-2020 academic year included a section on additional resources for learning to program in `R`, such as webpages associated with packages, and RStudio sponsored cheatsheets (link) and primers (link). Alongside these resources we also provide information about the Statistical Consulting and Research Services, so that attendees have a university resource for additional statistical services, and a point of contact if they encounter issues.

The third change made to the workshops offered during the 2019-2020 academic year, was adding a warm-up exercise at the beginning of each workshop. Per the feedback of the graduate student instructors who reflected that many workshop attendees would not engage with those around them, these warm-ups were added to increase group interactivity. Each warm-up consists of a display of workshop relevant example (e.g. data table, graph), which motivates the importance of the workshop content. Alongside the example, attendees are given a warm-up task to discuss with those around them. For example, in the warm-up for the *Data Visualization with `ggplot2`* workshop, the plot shown in Figure 3.4 is displayed and attendees are asked to "Describe the qualities of the plot" with those around them. With these warm-ups we have seen an increase the interactions of workshop attendees!

Finally, due to the large number of attendees from disciplines outside of the environmental sciences, we have begun to consider if it may be appropriate to provide

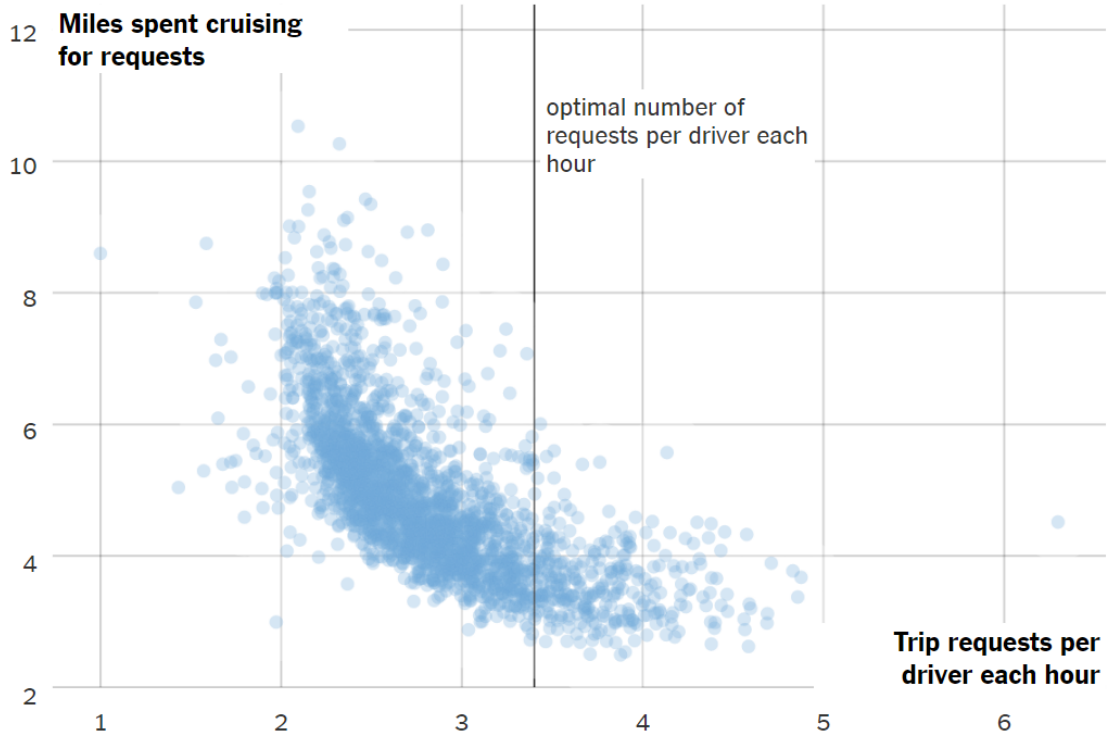


Figure 3.4: Warm-up for *Data Visualization with* `ggplot2` workshop, provided with a "Describe the qualities of the plot" prompt. By The New York Times; Source: Alejandro Henao and colleagues analyzing data from RideAustin.

a variety of data contexts for workshops. The possibility of data science workshops serving *every* scientific domain is discussed in more depth as a direction for future research. While there seems to be a possibility of more "general" data science workshops, attendees' responses to "what are you hoping to learn in this workshop?" provoked different potential directions for these workshops. A number of workshop attendees voiced that they were attending to gain skills outside of the domain of the workshop, such as learning to "analyze microbiom data with `R`," "geospatial analysis in `R`," or "how to integrate data into the `Ternary` package I'm learning." Each of these requests suggests the need for more specialized, discipline-specific workshops for working with data in `R`.

Sustainability of Workshops

To facilitate the sustainability of these workshops, we forged a partnership between our institution's library and the Department of Mathematical Science's Statistical Consulting and Research Services (SCRS). We believe a university's library is an optimal unit for offering these workshops, as it is both department-agnostic and a central hub for the entire university community. Furthermore, by partnering with a organization that provides statistical consulting, workshop participants are provided with a potential avenue if difficulties or additional questions arise—so the peer network is not shifted onto workshop instructors.

A data-engagement grant from the National Network of Libraries of Medicine during the 2018-2019 academic year supported the primary author in leading the workshops, becoming a Carpentries certified instructor, and incorporating the results of this research into the broader Data and Software Carpentry curricula. A $5,000 faculty excellence grant during the 2019-2020 academic year, facilitated the implementation of a "train-the-trainer" model, training two future graduate student instructors. Students were recruited from the masters and doctoral programs in statistics, but because of the widespread use of `R` across scientific fields, students from a variety of backgrounds hold the potential to be effective instructors. Both semesters, the authors met with these students for one hour a week to build students' facilities and confidence instructing each workshop. Each of these semesters, students taught different 30 to 45 minute portions of each workshop during, and acted as assistants for the remainder for the workshop.

The Carpentries does not require training for instructors to use their content, as The Carpentries materials are publicly available for use and adaptation (with acknowledgement). However, if the instructor or institution desires to advertise their workshops as Carpentries workshops there are two options: (1) the lead instructor is a Carpentries certified instructor, or (2) the institution requests a workshop through The Carpentries, who then recruits instructors for a fee. Through the primary author completing The Carpentries instructor

training, we were able to offer self-organized workshops interweaving the content from both Data and Software Carpentry workshops. Additionally, through this experience the primary author was able to guide the future workshop instructors through the process of becoming Carpentries certified instructors.

Similar to the Explorations in Statistics Research workshop model (Nolan and Temple Lang, 2015), the "standard" Carpentries workshop format takes place over an intensive two days. Self-organized workshops allow for the added flexibility of tailoring this format to be more conducive for busy students, faculty, and staff. This revised format has both benefits and costs. The additional time between each workshop helps to alleviate the brain fatigue often experienced in intensive workshops, and allows for participants to attend the workshops that are relevant to the skills they wish to acquire. However, in this extended format, workshops after *Introduction to* `R` are potentially considered "specialized" workshops and experience lower attendance. At an academic institution, there is the possibility of integrating this type of workshop series into a single credit course. When considering this as an option, however, institutions should think critically about how faculty and staff can still participate in these critical learning opportunities. Alternatively, institutions could offer undergraduate students the option of assisting in the implementation of these workshops for course credit, and allow for the possibility of students becoming lead or co-instructors as they progress through their program.

Limitations & Future Research

The sentiments heard by faculty members in this research, unearth the possibility that many faculty may be unaware of the computing skills necessary for their graduate students to participate in the entire data analysis cycle. Instead, students may have more relevant knowledge regarding the data science skills that are necessary for their research. Hence, the next iteration of this design work will be informed by the collection of the research (`R`) code produced by environmental science graduate students. Graduate students' research

code acts as artifacts of their research experience, providing "mute evidence" (Hodder, 1994) of the data science skills necessary throughout the data analysis cycle. The skills outlined by this research aid in reevaluating the content of these workshops, to ensure they cover the skills necessary for graduate-level environmental science research.

Additionally, the attendance of these tailored workshops by students, faculty, and staff from disciplines outside of the environmental sciences brings to question whether this type of tailored design work is necessary. Over a third of the workshop attendees came from disciplines outside of the environmental sciences, and, strikingly, these attendees reported similar workshop experiences to attendees from these targeted disciplines. This brings to question if there are common computational understandings necessary for research in *any* scientific field, which should be infused into *every* statistics and data science course. Alternatively, we saw a greater persistence across workshops by attendees from environmental science fields. This made us wonder, what are the drivers behind these individuals' continued attendance? Future research investigating the learning outcomes of workshop attendees holds the potential to provide fruitful insight on the necessity of discipline-specific learning opportunities.

Finally, despite the increasing availability of extracurricular workshops, research has yet to investigate the consistency or drift of these workshops. In this research, because of the large attendance at many *Introduction to* `R` workshops, a large number of questions would arise over the course of the afternoon. This lead to an inability to cover some of the workshop content in as much depth as hoped, yet some attendees remarked that "with so many people, [the workshop] had better discussions." A large scale analysis of the content covered by these workshops could unearth common questions or misunderstandings, aiding in the reconstruction of lessons to better scaffold learning.

Conclusion

Ten years ago, Nolan and Temple Lang declared that "modernizing the statistics curricula to include computing [...] is an issue that deserves widespread attention and action" (p. 106). Over the last ten years, we have seen both small (Revision Committee, 2014) and large (American Statistical Association Undergraduate Guidelines Workgroup, 2014) changes advocated to the statistics curriculum. Unfortunately, changes to graduate-level statistics service courses has received less attention and poses different issues.

Statistics courses that serve a variety of students (undergraduate, graduate, statistics major, non-major) reflect a snapshot of the statistics curriculum, but often act as many students' sole statistics course prior to conducting scientific research. Instructors of these courses thus grapple with difficult decisions of how they can ensure their students have both the statistical and "computational understanding, skills, and confidence needed to actively and wholeheartedly participate" in the scientific research arena (Nolan and Temple Lang, 2010, p. 106). For instructors unfamiliar with students' scientific disciplines, it can be difficult to "be bold and design curricula from scratch" (Nolan and Temple Lang, 2010, p. 106). The topics suggested by Nolan and Temple Lang (2010) represent a starting point toward building a taxonomy for computing in statistics for undergraduate and graduate statistics programs. These topics, however, may not be relevant to or emphasized by other scientific disciplines whose students enroll in graduate-level statistics service courses. In our research, we found that environmental science faculty stressed the importance of graduate students developing skills surrounding the fundamentals of working with data in `R`, software skills for data processing and preparation, creation of data visualizations, and usage of reproducible work flows.

The time is ripe for us to "update the foundational concepts and infrastructure" (He et al., 2019, p. 5) included in statistics service courses, in the new era of data science. As we work toward a more thorough integration of computing into these courses, this research offers a model for facilitating external workshops, which hold the potential to fill a critical

hole in the curriculum of many college programs. External workshops hold the opportunity for co-curricular learning, when paired with statistics service courses, so students leave their statistics service course with the computing skills necessary to engage in the entire data analysis cycle. Moreover, these workshops support university-wide data science literacy, facilitating avenues for faculty to acquire data science knowledge and skills which "they have not had the opportunity to learn well" (Nolan and Temple Lang, 2010, p. 106), and providing resources for instructors to meaningfully integrate discipline-specific computing skills into their classroom.

Acknowledgements

We would like to specially thank the participants from this study, without whom this research would not have been possible. We would also like to thank the workshop helpers for their time and assistance, helping to grow the data literacy across our campus. Lastly, we thank Mary Alice Carlson, Jennifer Green, Mark Greenwood, Megan Wickstrom, editor Johanna Hardin, and the reviewers for their insightful comments on this paper.

DATA SCIENCE SKILLS IN DATA-INTENSIVE ENVIRONMENTAL SCIENCE RESEARCH: THE CASE OF ALICIA AND ELLIE

Contribution of Authors and Co-Authors

Author: Allison Theobold

Contributions: Designed the study, collected the data, performed the analysis, interpreted the results, and wrote the manuscript.

Co-Author: Stacey Hancock

Contributions: Discussed data analysis, results, and implications, and edited earlier manuscripts.

Manuscript Information Page

Allison Theobold & Stacey Hancock

Harvard Data Science Review

Status of Manuscript:
__X__ Prepared for submission to a peer-reviewed journal
_____ Officially submitted to a peer-reviewed journal
_____ Accepted by a peer-reviewed journal
_____ Published in a peer-reviewed journal

Publisher: *Harvard Data Science Review*

Submitted:

Abstract

The importance of data science skills for modern scientific research cannot be understated. In today's changing data landscape, discussions surrounding broad sets of data science skills relevant to researchers have resonated from both environmental science and statistics education researchers alike. With statistics educators focusing on the skills necessary for future statisticians, and environmental science educators focusing on the skills needed for data-intensive environmental science research, neither discipline acknowledges the necessity of data science skills for scientific *practitioners* of statistics—those that are at the intersection of these two fields. This space, however, is navigated by nearly every graduate student in the environmental sciences, as they conduct their independent research.

The focus of this paper is an in-depth examination of the data science skills employed by two environmental science graduate students throughout their research. Students' research code was collected as they progressed through their graduate program and qualitatively analyzed for emergent themes of data science skills, mapping the evolution of these skills over time. This case study, rather than focusing on broad sets of computing skills potentially relevant to environmental science researchers, offers a compelling example of specific data science skills necessary for students to engage in data-intensive environmental science research. The results present the data science skills used by Alicia and Ellie, their experiences acquiring these necessary skills, and a discussion of the impact of student preparation and support on their acquisition of the data science skills necessary for their research.

Introduction

The ripples of today's data revolution can be felt across nearly every scientific field. This revolution is not only about big data, but about all sizes and types of data, and the tools researchers use to work with them. Amidst the changing data landscape, environmental

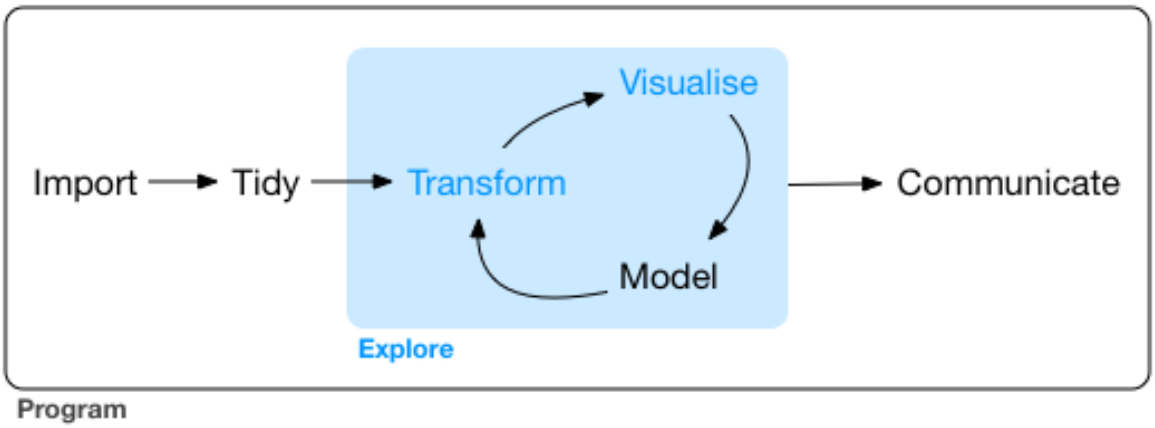


Figure 4.1: Data Analysis Cycle, Wickham, H. & Grolemund, G. (2017) *R for Data Science*. Sebastopol, California: O'Reilly. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License.

science and statistics educators have outlined the data science skills necessary for researchers in each respective field (Hampton et al., 2017; Nolan and Temple Lang, 2010). However, these conversations fail to acknowledge the importance of statistics to scientific research and the necessity of data science skills as researchers participate in the data analysis process. Hence, we have little knowledge about the data science skills necessary for environmental science *practitioners* of statistics to successfully conduct their research.

For this study, the "data analysis cycle" is considered to consist of the stages from data importation to the communication of results, where data modeling is only a single component. This definition aligns with the "data science" cycle outlined by Wickham and Grolemund (2017), as seen in Figure 4.1. This cycle speaks to the variety of "data science skills" that may be necessary throughout one's data analysis. While components of these data science skills may include general programming concepts such as loops, user-defined functions, or conditional statements, but the focus of data science skills differ fundamentally from general programming skills. Rather than focusing on computer architecture, design, and application, data science skills focus on the data.

Consider an environmental science graduate student conducting research for their master's or doctoral degree. Because of the importance of statistics to environmental science

research, the student has completed graduate-level statistics coursework. Yet, when setting out to analyze the data they've collected over the last two years, they face hurdles in cleaning, wrangling, and visualizing their data before they even begin their statistical analysis. Although "data analysis is at the heart of data science," (Wing, 2019), the data science skills necessary for data-intensive environmental science research start long before a statistical model is fit; yet, we lack an understanding of the key data science skills environmental science graduate students employ throughout the entire data analysis cycle (Wickham and Grolemund, 2017). This case study focuses on an in-depth investigation of the data science skills which two environmental science graduate students, Alicia and Ellie (pseudonyms), as they use statistics throughout their research.

Over the last 20 years, statistics has grown to be considered vital to environmental science research. In fact, today "it would be extremely difficult to publish a manuscript in ecology without any statistical testing" (Hampton et al., 2017, p. 547). Thus, statistics preparation has been readily incorporated into the environmental science graduate curriculum. Yet, environmental science educators (Hampton et al., 2017; Teal et al., 2015) claim that the environmental science curriculum continues to be devoid of any formal training in computing. This lack of formal training potentially leaves environmental science faculty to assume that students are able to analyze their data because they've taken a statistics course.

Because of this importance of statistical training to graduate students in these fields, the women at the focus of this study were recruited from a first semester graduate-level applied statistics course. They were then followed through their graduate program, with routine collection of the `R` code they generated for their research. A student's research code acts as artifacts of their research experience, providing "mute evidence" (Hodder, 1994) of the data science skills necessary for their participation in data-intensive environmental science research. Furthermore, these artifacts, unlike the spoken word, endure physically (Hodder, 1994), reflect the results of "actual behavior rather than reported approximations"

(Merriam, 2009, p. 148), are nonreactive and non-obtrusive, and "can be used over long time periods as longitudinal monitoring devices" (Merriam, 2009, p. 149).

Though often "Environmental Science" refers to a specific discipline in the literature, for this paper we refer to the collection of fields who perform research in the biological and environmental sciences as "environmental science." At our institution, these are the fields whose students are highly recommended to enroll in graduate-level applied statistics coursework, and include departments of Ecology, Land Resources and Environmental Sciences, Plant Sciences and Plant Pathology, Animal and Range Sciences, and Earth Sciences. In the context of student's independent research, "reproduciblity" does not refer to the replication of a study. Instead, "reproducibilty" refers to another researcher's ability to recreated the results of a project, "given only a set of files and written instructions" (Kitzes, Turek, Deniz, 2018). In this paper, we discuss reproducibility both from the habits perspective, and the technologies perspective (Sandve et al., 2013).

This paper outlines the themes of data science skills used by Alicia and Ellie (pseudonyms) throughout their data analysis cycle, and how their skills evolved over time. Alicia came to graduate school from a Bachelor's degree in the social sciences. Over the course of her graduate program, Alicia enrolled in a single statistics course, and no field-specific quantitative methods courses. Furthermore, for the research required for her degree, Alicia received little to no statistical or computational support from her adviser. In contrast, Ellie came to graduate school after completing a Bachelor's degree in engineering, took numerous statistics courses and quantitative methods courses specific to her field, and was mentored by an adviser who supported her statistically and computationally. It is because of these substantially different experiences that this paper focuses on Ellie and Alicia, contrasting the data science skills used by each student throughout their research. This case study, rather than focusing on broad sets of data science skills that could potentially be relevant to graduate students in the environmental sciences, serves instead as a rich example of the data science skills necessary for environmental science graduate students to

engage in the data analysis cycle.

Data Science in Statistics & the Environmental Sciences

Over the last 10 years, we have seen substantial growth in research focusing on the data science skills necessary for data-intensive research, spanning a variety of scientific fields. Literature in the environmental sciences has primarily focused on two areas: (1) outlining the computational ill-preparation of graduate students by their curriculum, and (2) describing broad computational skills necessary for participation in data-intensive environmental science research. Although literature in these fields outline researchers' need for computational skills for working with, visualizing, and analyzing data, none of these conversations acknowledge the role a student's statistics education functions in their attainment of these necessary skills. During this time period, statistics educators transitioned from questioning to what extent computing should be included in the statistics curriculum to including computing as an integral part of the curriculum. This shift was reinforced by (1) changes in the technology for implementing statistics, (2) revisions to the guidelines for undergraduate programs in Statistics, and (3) the beginnings of data science in the statistics curriculum.

In this section, we discuss the current climate of data science in the fields of Statistics and the environmental sciences. First, we detail the research outlining the computational training of environmental science graduate students and the literature on the computational skills necessary for data-intensive environmental science research. Next, we outline the statistics education research which elevated the importance of computing, advances in the tools available for working with data, and how these changes are reflected in today's statistics curriculum. Finally, we conclude with a discussion of the barriers educators potentially face when incorporating data science topics into the curriculum.

Data Science in the Environmental Sciences

The conversation surrounding the gap between the data science skills possessed by graduate students in the biological sciences and the skills necessary for biological research was initiated by Andelman and colleagues (2004). Through a graduate-level seminar course, the authors learned that students were unprepared with both the computational and statistical skills necessary for data analysis. Disappointingly, they found that "ninety-three percent of students did not have skills in the scripted programming languages (e.g., SAS or MATLAB) that are needed for the integration of large data sets" (p. 244). An outpouring of research on the importance of computational preparation for undergraduate and graduate students in scientific fields followed (Dodds et al., 2007, 2008; Eglen, 2009; Green et al., 2005; Hastings et al., 2005; Kelling et al., 2009; Wilson, 2006; Wilson et al., 2008; Wing, 2006).

Strasser and Hampton (2012) furthered this conversation, with a study of the incorporation of data management topics into undergraduate ecology courses. In surveys of instructors likely to be teaching future graduate students in ecology, these researchers found that less than 20% of instructors included key data management topics, such as workflows, databases, and reproducibility in their courses. The importance of these skills was affirmed by a large percentage of the ecology instructors (77%) stating that "data management should be taught in a different course" (p. 10). Yet, these results suggest that—across institutions—"data management education is not currently a priority for ecology instructors" (p. 10). Later that same year, an environmental science graduate student led a large-scale study of the "technological and computational experience of environmental scientists" (Hernandez et al., 2012, p. 1068). In a survey of graduate students in the environmental sciences, the authors found that over 74% of the students surveyed reported they had no skills in any programming language—including `R`—and only 17% reported basic skill levels in any programming language. Hence, students are leaving their graduate programs without the computational skills necessary for data-intensive research in these fields.

As the literature on the computational ill-preparation of graduate students in scientific fields amassed, a cohort of scientific researchers began the conversation around what computing practices every researcher could adopt, "regardless of their current level of computational skill" (Wilson et al., 2008, p. 1). These researchers outlined a "minimum set of tools and techniques that [they] believe every researcher can and should adopt" (p. 20). They distilled these skills into six topics: (1) data management, (2) software, (3) collaboration, (4) project organization, (5) tracking changes, and (6) manuscripts. The first two topics, data management and software, reiterate ideas previously heard in the literature, emphasizing the importance of reproducibility. However, emphasizing researchers' abilities to make project and manuscript collaboration easy and to record how a project changes over time, were skills not previously discussed in the environmental science literature.

Adding to this conversation, educators and researchers from a variety of environmental science fields gathered to write a formative piece outlining the skills and knowledge necessary for data-intensive environmental science research (Hampton et al., 2017). Where Wilson and colleagues outlined the bare minimum techniques scientific researchers should employ in their research, Hampton et al. add an environmental science perspective, outlining the "skillset required by environmental scientists to succeed in the kind of data-intensive scientific collaboration that is increasingly valued" (p. 548). The authors distinguish five broad classes of skills: (1) data management and processing, (2) analysis, (3) software skills for science, (4) visualization, and (5) communication methods for collaboration and dissemination. The broad skillset outlined by these authors, however, is not tailored to any specific population of researchers, leading the reader to wonder which computing skills are necessary throughout *every* environmental science field, and which are irrelevant or purely tangential to other areas.

Across all of these conversations, however, attention has yet to be paid to the substantial role students' statistics education potentially plays in their attainment of the data science skills necessary for research. Andelman and colleagues (2004) lamented that,

in addition to students' unfamiliarity with scripted programming languages, they were also "unfamiliar with multivariate statistics and with the range of models for regression and analysis of variance" (p. 244). Strasser and Hampton (2012) found that the most common courses faculty voiced as potentially covering the "data-related topics" included in their survey were "ecology laboratories, advanced ecology courses, or statistics courses" (p. 7-8). In the survey given to environmental science graduate students across California, Hernandez et al. (2012) included spatial or time series analysis courses as courses that relate "to the management and analysis of large or complex data" (p. 1070), and students' level of proficiency with `R`.

Data Science in Statistics

The discussion of computing in Statistics dates back to 1962, when Tukey stated that "in the future [Statistics] can and should contribute much more" to data analysis (p. 3). This call for the field of statistics to "seek out novelty" through adopting computational methods was reiterated over the subsequent decades by both statisticians (Breiman, 2001; Friedman, 2001) and statistics educators (Biehler, 1997; Cleveland, 2001; Higgins, 1999; Moore et al., 1995; National Research Council, 1994; Nolan and Speed, 1999) alike. During the early part of this century, statistics educators voiced concerns that the traditional statistics curriculum, with its mathematical foundations, was not keeping up with the computational and graphical tools being used by professional statisticians (Biehler, 1997; Brown and Kass, 2009; Moore et al., 1995; Peck and Chance, 2005). Calls for change resonated nationally, with statistics educators outlining new guidelines for undergraduate majors and minors in Statistics (Bryce et al., 2001; Cannon et al., 2002) and methods for incorporating computing into mathematical statistics courses (Higgins, 1999; Horton et al., 2004; Nolan and Speed, 1999; Reid et al., 2003).

In 2010, Nolan and Temple Lang shifted the paradigm of these conversations by painting a broad picture of the computing skills with which successful statisticians must be facile. Rather than questioning the extent to which computing should be incorporated

into the undergraduate statistics program, the authors argued that these key computing skills should be wholeheartedly incorporated throughout every course in the statistics curriculum. Nolan and Temple Lang lamented that the current statistics curriculum was not preparing students with the computational proficiency or the "confidence needed to overcome computational challenges" (p. 97). Instead, students were being "told to learn how to program by themselves, from each other, or from their teaching assistant in a two-week 'crash course' in basic syntax at the start of a course" (p. 100). Yet, they had found that graduates of their bachelor's and master's programs "spend much of their efforts retrieving, filtering, and cleaning data and doing initial exploratory data analysis" (p. 99), skills with which they did not have after leaving the classroom.

Amidst these conversations, packages were being created which would fundamentally change how users interact with `R`. In 2005, the `ggplot2` package was created to produce statistical, or data, graphics; but unlike most other graphics packages, it had the "deep underlying grammar" of Leland Wilkinson's *Grammar of Graphics* (Wickham, 2016, p. 1; Wilkinson, 2005). Over the next ten years, Wickham and colleagues would produce a suite of `R` packages to facilitate a "clean data science workflow" (Wickham, 2017), universally known as the `tidyverse` (Wickham, 2017), including `stringr` (2009), `dplyr` (2014), `RMarkdown` (2014), `tidyr` (2014), `readr` (2015), `purrr` (2015), `tibble` (2016), and `forcats` (2016). The creation of these packages brings to fruition Biehler's desire for the field of Statistics to "produce software more adequate both for learning and doing statistics" (1997, p. 167), and the aspirations of Moore et al. that these technological advances "may at last bring widespread change to college teaching" (1995, p. 250).

National calls for transforming undergraduate statistics education and a dramatic change in statistical computing technology pressured the 2014 American Statistical Association (ASA) President, Nathaniel Schenker, to convene a workgroup to update the association's guidelines for undergraduate programs in Statistics. These new guidelines included an increased emphasis on data science skills; specifically, students' ability to

"access and manipulate data in various ways, use a variety of computational approaches to extract meaning from data, and program in higher-level languages" (American Statistical Association Undergraduate Guidelines Workgroup, 2014, p. 7). While these changes reflected the growing consensus of the importance of incorporating computing throughout the statistics curriculum, the preponderance of statistics education literature up to that point had focused on the introductory and mathematical statistics courses. This led *The American Statistician* to produce a special issue on "Statistics and the Undergraduate Curriculum" in 2015. The issue encouraged submissions of broader issues throughout the statistics curriculum, publishing the first statistics education perspectives on infusing "data science" topics throughout the undergraduate statistics curriculum (Baumer, 2015; Grimshaw, 2015; Hardin et al., 2015; Nolan and Temple Lang, 2015). In the same issue, George Cobb charged statistics educators to rebuild the undergraduate statistics curriculum from the ground up, as "what we teach lags decades behind what we practice" and "the gap between our half-century-old curriculum and our contemporary statistical practice continues to widen" (p. 268). Cobb argued that while computer scientists teach algorithmic approaches at the elementary level, statisticians do not but should. Moreover, statistics educators need to think deeply about incorporating computing meaningfully into the classroom. Computing deserves more attention in the statistics curriculum than simply inserting "a single new 'big data' unit into an existing course" or inserting "a new computing course into the existing curriculum" (p. 275).

Despite these calls for innovating the statistics curriculum, we continue to see a mere 60% of graduate students in environmental science fields reporting leaving their program with a "basic skill level in statistical applications, including `R`" (Hernandez et al., 2012, p. 1069), regardless of their enrollment in statistics courses. This disparity between the importance of computing voiced by statistics educators and the computing skills with which environmental science students report leaving their program suggests computing continues to be absent from many statistics courses, especially service courses. In this year's special

issue on the current state of "Computing in the Statistics Curriculum," celebrating the ten-year anniversary of "Computing in the Statistics Curriculum" (Nolan and Temple Lang, 2010), the *Journal of Statistics Education* is encouraging these types of conversations.

Barriers to Incorporating Data Science in the Curriculum

Although calls for incorporating computing throughout the statistics and environmental science curricula have resonated for the last ten years (Jones et al., 2006; Joppa et al., 2013; Laney et al., 2015; Manyika et al., 2011; Mokany et al., 2016; Peters and Okin, 2017; Smith, 2015; Teal et al., 2015), we continue to see studies reporting the computational ill-preparation of environmental science undergraduate and graduate students by their curriculum (Hampton et al., 2017; Teal et al., 2015). This bring us to question why these skills are still so rare when the need for them is now widely recognized?

Almost ten years ago, 71% of the ecology instructors surveyed by Strasser and Hampton (2012) reported barriers to including data management topics in their course(s). These barriers included the instructor's lack of time or their lack of knowledge in the topics, students' lack of the necessary quantitative skills, or a lack of alignment of the topics with the instructor's course. These obstacles can be distilled into "attempting to fit more material into already-full courses and curriculum, which are taught by people who do not feel prepared to address topics relevant to big data and data-intensive research" (Hampton et al., 2017, p. 547). Disappointingly, this lack of computational training necessary for data-intensive environmental science research hampers the progression of scientific research and begets numerous hidden consequences.

Teal and colleagues contend that "most researchers learn what they know about programming and data management on their own or the information is passed down within a lab" (2015, p. 136). This do-it-yourself approach results in students potentially "picking up bad habits, misunderstandings, and, more importantly, the wrong concepts" (Nolan and Temple Lang, 2010, p. 100). Students' initial knowledge shapes the way in which they address tasks, making some tasks impossible. Additionally, students can spend weeks or

months doing things that could be done in hours or days, learning just enough to get them through, without learning the simple ways to do things or abstracting what they learned to broader classes of tasks. Furthermore, students may be unaware of the reliability of their results, and may often be unable to reproduce their work.

Clearly, the current situation is unsatisfactory; however, with the growing presence of big data and continuing advances in technologies, the situation may be difficult to remedy as statistics and environmental science instructors may continue to experience barriers to incorporating computing into their courses (Strasser and Hampton, 2012). Statistics educators have outlined broad sets of computing skills which ought to be infused into undergraduate programs in Statistics (American Statistical Association Undergraduate Guidelines Workgroup, 2014; Nolan and Temple Lang, 2010) and what knowledge statistically literate citizens should possess (Gould, 2010; Moreno, 2002; Utts, 2003). Yet, no discussions have outlined the space in-between—the computing skills necessary for scientific practitioners of statistics. Similarly, many skills in the taxonomy outlined by Hampton et al. may be irrelevant or purely tangential for environmental science graduate students conducting data-intensive research. This incomplete information around the computing skills necessary to implement statistics in graduate-level environmental science research makes it difficult for instructors of statistics courses that serve these students to "be bold and design curricula from scratch" (Nolan and Temple Lang, 2010, p. 106).

Learning from Student-Generated Code

Even with the elevated importance of computing in Statistics and the environmental sciences, there exists a dearth of literature investigating how students learn the computing skills necessary for research in their respective field. Information can be borrowed, however, from computer science educators, who have investigated what can be understood about student learning through artifacts of their programming. Over the last ten years, computer science educators have expended a large amount of effort researching novice computer programmers through three main avenues: (1) identifying difficult concepts in the

Introductory Computer Science (CS-1) curriculum, (2) mapping the evolution of students' programming, and (3) examining compilation errors.

Researchers investigating the concepts students in CS-1 courses find the most difficult typically present students with a set of questions targeting a specific concept and then record the proportion of correct responses (Caceffo et al., 2016; Cherenkova et al., 2014; Lahtinen et al., 2005; Milne and Rowe, 2002). To map the evolution of students' learning, researchers have focused on analyzing the differences between each subsequent save of a program (Bilkstein, 2011; Worsley and Blikstein, 2013). Compilation errors have been examined with two different analysis strategies. First, researchers have used static analysis tools to explore the major programming mistakes novice programmers make (Altadmri and Brown, 2015; Brown and Altadmri, 2014; Hristova et al., 2003). Alternatively, researchers have probed how interactive environments, with error checking, impact students' learning (Bulmer et al., 2018; Brown and Altadmri, 2014).

Although each of these investigative methods illuminate different aspects of students' code, computer science education has yet to employ a method of investigation which can shed light on the specific skills students use for a given problem or task. This type of inquiry necessitates a qualitative method of investigation. Qualitatively analyzing students' code, allows us to holistically describe the skills students use throughout their code, organize these descriptions into themes of skills employed by each student, and compare these emergent themes across students and across time.

Methods

This article details an embedded, comparative case study (Yin, 2009) of the data science skills necessary when implementing statistics in data-intensive environmental science research. Comparative case studies allow for the analysis of contrasting cases, promoting a deeper understanding of a single-case finding, "grounding it by specifying how and where and why it carries on as it does" (Miles and Huberman, 1994, p. 29). The environmental

science graduate students, Alicia and Ellie (pseudonyms), were the cases, and the research code produced for their respective degrees were the embedded units of analysis. Looking at the research code produced by Alicia and Ellie does not assess the prevalence of the phenomenon of data science skills necessary for data-intensive research (Yin, 2009, p. 56). Rather, this study provides insight into the critical data science skills necessary for graduate-level, data-intensive environmental science research (Stake, 2006, p. 5-6).

Students' `R` code was selected as the embedded units of analysis because of the growing use of `R` across environmental science disciplines. In these fields, `R` has grown to be the "primary tool reported in data analysis," increasing from 11.4% in 2008 to 58% in 2017 (Lai et al., 2019, p. 1). This change in the programming landscape is due to a variety of reasons, including, but not limited to, `R`'s availability as free software (R Core Team, 2020), the capabilities of over 10,000 user-created packages for `R`, and the creation of the RStudio integrated development environment for working in `R` (RStudio Team, 2015b). `R` is an interpreted language, where the user creates "objects," such as vectors, `data.frame`s, matrices, or lists, and can use these objects in the default functions loaded into `R` ("base `R`"). To extend the capabilities of `R`, the user can load user-created packages, such as the `tidyverse` (Wickham, 2017).

Participants & Data Collection

At our university, the first-semester of a two-semester graduate-level Applied Statistics course sequence (GLAS I & II) services graduate students from a variety of scientific fields, including, but not limited to, the environmental sciences. GLAS I requires that graduate students have taken an undergraduate introductory statistics course. For students continuing on to additional statistics courses, GLAS I serves as the required prerequisite course.

Students were recruited from GLAS I in the spring of 2018. As the intention of the study was to describe the computational abilities necessary for data-intensive research across environmental science fields, every graduate student from environmental science fields taking

the course for their respective master's or doctoral programs was considered. Every student satisfying these criteria was requested to complete a survey detailing their departmental affiliation(s), their program of study, the degree they are seeking, and their year in their program. The nine students who completed the survey were asked to participate in a study whose aim was to understand the computational abilities necessary when applying statistics for research in their field, and how they gain these necessary computational skills. All nine students agreed to participate in a one-hour meeting at the beginning of each semester over the next three semesters, where they would provide any `R` code they generated for their research and outline where they had acquired the computational skills they used throughout their code.

In the fall of 2018, each of these nine students was contacted for an interview. At this time, one student did not respond and two students stated they had not yet generated any `R` code for their research. The six remaining students participated in a one-hour interview, as described previously. In the spring of 2019, these six students were contacted again for an interview. At this time, one student did not respond. However, one of the two students who had not produced research code in the fall stated they had produced research code, and agreed to participate in an interview. Finally, in the fall of 2019, the six students interviewed in the spring of 2018 were contacted for a final interview, as well as the student who did not participate in an interview the previous spring. For the final interview period, one student was unavailable due to spending the semester at a research site, and one student had not produced any new research code. Thus, in the fall of 2019, four students were interviewed.

One week prior to each interview, each student was requested to submit any new or modified research code they had produced since the last interview. Analytic memos for each of these code files were produced by the primary author to synthesize the data science skills used throughout the code into higher-level analytic meanings (Miles et al., 2014, p. 95). For research code that had been previously submitted, the modifications were inspected for additional data science skills not seen in the previous version(s) of the code. During each

semi-structured interview (Bernard, 1988), students were asked where and how they had acquired each of the data science skills identified in the analytic memos of their research code. If their response warranted further investigation, follow-up questions were asked to illuminate why the participant chose this resource when acquiring the computational skill in question. For example, if a student voiced acquiring the ability to use `apply` statements in a course they took, further information was sought regarding in what course the skill was learned. If a student voiced the Internet as their resource in acquiring the ability to use conditional statements, additional information was sought regarding the specific Internet resources they used. The full interview protocol is included as an Appendix. Through this process, it became clear that two students, Ellie and Alicia, stood out as examples of two extremes of the phenomena under study (Yin, 2009).

The rationale for selecting Ellie and Alicia was two-fold. First, the experiences of these women represent two opposing extreme points in the computational preparation and support of environmental science graduate students. Their experiences differed in four primary ways: (1) Ellie had extensive programming experience during her Bachelor's degree, whereas Alicia had only slight exposure to programming languages during her Bachelor's degree; (2) Ellie was completing a Graduate Certificate in Statistics, and Alicia was discouraged by her adviser from enrolling in additional statistics courses after GLAS I; (3) Ellie enrolled in many quantitative methods courses in her field, such as Environmental Data Management, with Alicia enrolling in no field-specific quantitative methods courses; and (4) Ellie had an adviser and a committee member with extensive computing backgrounds who valued and supported extensive training in statistics and computing, but Alicia's adviser provided little to no statistical or computational support and from Alicia's perspective did not value formal training in statistics or computing.

Second, the research code produced by Ellie and Alicia also represented two extremes of the types of computational tasks environmental science researchers might face. Where a large amount of work done in Ellie's lab and for her research projects focused on creating

and maintaining `R` packages, the majority of computational tasks Alicia faced were related to wrangling, visualizing, and modeling her data. Therefore, the two cases described in this study represent two comparative cases of the data science skills necessary for participation in data-intensive environmental science research.

Data Analysis

We engaged in a case-oriented, four-stage analysis, led by the primary author. In the first stage, the `R` code produced by each participant was read through, and short, descriptive codes were given to each statement of code. During this process, the previous analytic memos created for the `R` script were referenced, however, these memos capture the data science skills used throughout the script. The descriptive codes, instead, detail the actions being taken in each statement of code. A "statement" of code is a line, or set of lines, which constitute a syntactic unit. Statements were considered to be the smallest syntactically valid statement of code; some statements were completed in a single line of code, while others took multiple lines. The following lines of code provide an example of two types of statements, where the statement shown in Figure 4.2 is defined by an assignment (`<-`) operation, and the statement shown in Figure 4.3 is defined by a `function` definition operation.

```
predictions1 <- matrix(nrow = totalRealizations,
                       ncol = length(time))
```

Figure 4.2: Statement 1, Data Analysis Example

During the descriptive coding process, a description was given to each code statement. These descriptions summarized the action the code statement performed. For example, the code statement in Figure 4.2 was described as "create a matrix using a built-in function, name arguments, use previously defined variables as inputs." The statement in Figure 4.3 was coded as "create a user-defined function which: takes multiple inputs, filters a vector using brackets, mutates an existing variable using a built-in function to create a new

```
nmle <- function(P, t, y, N15_NO3_0){
  yhat <- N15_NO3_0 * (1-exp(-P[1]*t))
  -sum(dnorm(y,yhat,exp(P[2]), log = T))
}
```

Figure 4.3: Statement 2, Data Analysis Example

variable, uses function inputs and previously defined variables as inputs for calculation which uses a built-in function and named arguments, returns the calculated value." During the descriptive coding process, the primary author did not reference themes of computing skills previously outlined in the literature. Aside from the verbiage used to describe common data manipulation actions (e.g., filter, select, mutate) (Wickham et al., 2018), the descriptions assigned to each code statement were allowed to naturally emerge. As eluded in the description of the first code statement, efficiency was an additional consideration that was made during the descriptive coding process. For this consideration, each code statement was inspected to see (1) if it was the same as another code statement seen elsewhere in the code, and (2) if the statement referenced previous code statement(s).

Additionally, during this first stage, the interviews for each participant were transcribed verbatim. The primary author then read the transcripts looking for references to the paths through which the participants voiced having acquired the data science skills used in their code. These segments were then descriptively coded to reflect the paths the participants used when acquiring the data science skill(s) in question. The interviews with each participant provide data triangulation (Denzin, 1978; Patton, 2002) in the understanding of the data science skills used by each participant in their research code.

After descriptive codes were created for each participant's `R` scripts and interviews, the primary author began the second stage of pattern coding. Pattern coding groups the descriptive codes into a smaller number of themes (Miles et al., 2014), an analog to quantitative methods for identify clusters or factors of a set of variables. First, at each

time point, each participant's R scripts were inspected for emergent themes of data science skills. At each time point, categories of data science skills that existed throughout each participant's code were retained. Once these themes were outlined, a concept map was created to visualize the interconnected nature of the themes. For example, throughout Alicia's code, there were a number of different methods used for **data wrangling**, such as filtering rows from a `data.frame`, as seen in Figure 4.4 below. However, each statement requires a different type of understanding of **data structures** in R, another emergent theme.

```
PADataNoOutlier[which(PADataNoOutlier$`Fork Length`)]

subset(RPMA2GrowthSub, StockYear < 2004)

RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 3]
```

Figure 4.4: Three statements from Alicia's code, displaying the intersection of the themes of **data wrangling** and **data structures**.

The pattern codes found in stage two laid the groundwork for the cross-unit analysis of participants' R code over time. During this third stage, a master concept map was created for each participant, summarizing how their emergent themes changed over time. For example, in the fall of 2018, Alicia's R scripts showed no record of how the data she was analyzing were manipulated, but in the spring of 2019, these **data wrangling** skills began to appear in her R scripts.

The final stage consisted of a cross-case analysis of the emergent themes of each participant. While the themes for each participant may differ, a cross-case analysis allows for "a deeper understanding of the themes across many cases"—it illuminates how these themes "are qualified by local conditions," and thus develops "more sophisticated descriptions and more powerful explanations" (Miles et al., 2014, p. 101). Following this final coding process, both authors met to discuss the coding procedure and scrutinize the themes assigned to each coding task. Ultimately, a consensus was reached regarding the within-case themes

and cross-case comparisons.

Results

The concept maps of Alicia and Ellie's data science skills speak to the entwined nature of each student's data science skills, and create a trajectory of how each student's data science skills evolved over time. Examples of data science themes and their associated skills are outlined in Table 4.1. The skills are not exhaustive, but paint a picture of the types of data science skills that are associated with each emergent theme. Unsurprisingly, we see an alignment between the themes of data science skills which emerged from Alicia and Ellie's code and the stages of the data science cycle (Wickham and Grolemund, 2017). The themes of **data wrangling**, **data visualization**, and **data model** see a direct overlap with the "explore" stage of this cycle, while **workflow**, **`R` environment**, and **data structures** address the nature of data science skills that may be necessary throughout the entire cycle.

Theme	Skills Included
Data Model	`lm`, `nls`
Data Structures	`data.frame`, vector, `list`
Data Visualization	scatterplot, density plot, group colors
Data Wrangling	filtering rows, selecting columns, mutating variables
`R` Environment	data access, user-defined functions, vectorization
Workflow	readability, reproducibility, efficiency

Table 4.1: Themes of data science skills seen in Alicia and Ellie's research code, alongside examples of their associated skills. The skills associated with the theme of data model are described by their associated `R` functions, where `lm` references a linear model and `nls` references a non-linear least squares model.

In this research, we did not investigate the appropriateness of the analysis strategies pursued by each student; instead, we identified the type of statistical model they employed

and any computational skills associated with that modeling strategy. Furthermore, we only had access to the research code each participant provided; thus, we did not identify students' use of data science skills not included in their R code. In the following sections, we outline the themes of data science skills used by each student at each time point of the study, with attention paid to mapping the intertwined relationship of these themes, and how themes evolved over time.

Alicia

The spring of 2018 was Alicia's first semester as a graduate student, pursuing a Master's degree in Ecology. She had completed a Bachelor's in a social science field at a large research university in the western United States directly before beginning graduate school. Alicia had minimal programming experiences during her Bachelor's degree, with experiences stemming through three main outlets: helping a postdoc in her lab analyze data from a lab trip exposed her to R, completing the Calculus series for engineers exposed her to MATLAB, and enrolling in an information technology course exposed her to Python and Java. In her first semester as a graduate student, Alicia enrolled in GLAS I at the recommendation of her adviser. Despite requesting additional statistical preparation, however, Alicia would complete only that single statistics course during her master's degree. Furthermore, as Alicia did not enroll in additional quantitative methods courses specific to ecological research or participate in any university-sponsored or external R workshops, GLAS I also served as the only quantitative course Alicia completed during her master's degree. The direction of Alicia's program of study was largely dependent on her adviser, whom Alicia described as not valuing formal training in statistics or computing. Moreover, during three semesters of conversations, it became clear that Alicia's research was largely done with little to no statistical or computational support from her adviser.

Over the duration of the study, Alicia produced and revised two R script files.

Fall 2018 In her second semester as an Ecology graduate student, in the fall of 2018, Alicia had begun the preliminary analysis of her data. Thus far in her research, Alicia had generated one `R` script, consisting of repeating the same process six distinct times, a process familiar to her from having taken GLAS I in the spring. The process was as follows: (1) use `lm()` to fit a simple linear regression, (2) use `plot()` to create a scatterplot of the variables included in the regression, (3) use `abline()` to add a linear trend line from `lm()` to the scatterplot, and (4) inspect the statistical model using `summary()`. Although this process appears simple, Alicia's `R` code provides insights into the data science skills early graduate students may require. Figure 4.5 displays the themes of data science skills Alicia used throughout her `R` script. Themes are displayed in bold in boxes, and specific skills associated with each theme are included in each ellipse, where skills belonging to multiple themes are included in the intersections.

When inspecting Alicia's `R` script, one immediately notices a lack of workflow and organization. There are no statements of code reading in the data, because Alicia was unsure if she would get different results if she wrote a line of code to import the data rather than using the "Import Dataset" button in RStudio. Alicia's poor workflow extended throughout the data science skills seen in her `R` script. Accordingly, the theme of **workflow** lies at the center of the collection of data science skills Alicia's used throughout her code.

Alicia used code comments, sparingly, to indicate which variables were being used for the different analyses. These comments, however, outside of object names, were the only indication that more than one dataset was being used for analysis. A code comment reading "`#OUTLIER REMOVED`" indicated a transition from the previous data being analyzed, with no `R` code to explain the differences between the two datasets. When questioned about how she removed the "outlier" from her data, Alicia stated,

> I know that I could subset it, I just forgot how. And I was trying to crunch this out for my proposal, so I just wanted to get the data out. So, I went into Excel and deleted it and ran it again. I could just subset it, and it makes its own dataset in `R`, but I wanted to be able to look at them in Excel, too. And compare them to each other.

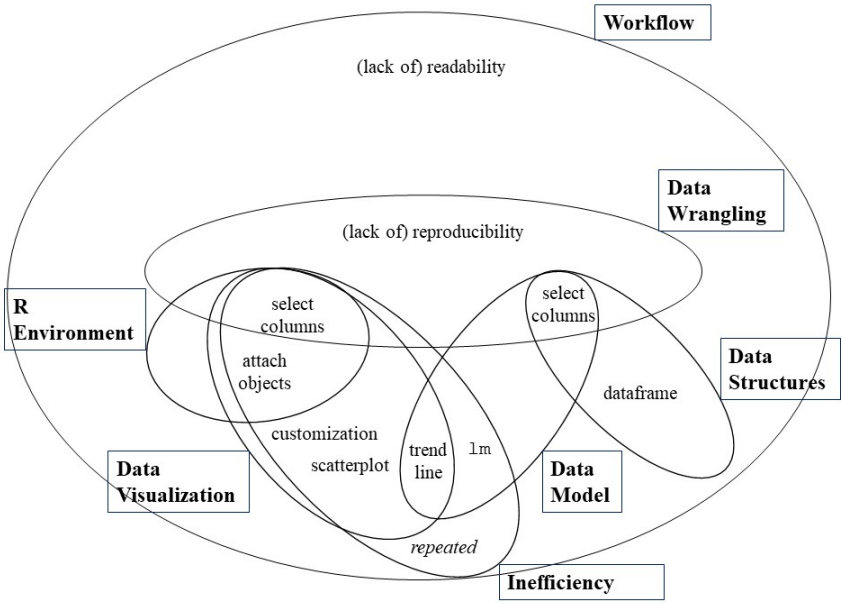


Figure 4.5: Concept map of data science skills seen in Alicia's `R` script in the fall of 2018. Themes are loosely arranged into groups, with theme names described in boxes. The list of specific skills seen within each theme are included in the ellipses. The skills included are not exhaustive, but rather are indicative of the type of skill seen in the research code. Many skills fall into multiple themes, as evidenced by the overlapping ellipses and the appearance of the same skill in multiple locations.

Alicia's lack of scripted, reproducible data manipulations lies at the heart of the **data wrangling** theme in Alicia's concept map, as her actions of mutating variables and filtering rows were not seen in Alicia's `R` code, but rather in her recount of the data manipulations she carried out in Excel.

This brute force non-reproducible, approach was found throughout Alicia's code. When questioned about her repeated process of modeling the data, visualizing the variables in the model, and using the data model to add a trend line to the visualization, Alicia reflected that the process she wanted to carry out was different from what she remembered learning in her statistics course. So, she had to use "different code than what [her instructor] gave [us] to run." She then said, "I just found something online that kind of looked like what I wanted, so I put that code in there." The theme of **inefficiency** captures Alicia's

repeated process of copying, pasting, and modifying this same statement multiple times. This theme conforms to Alicia's process of finding code online that addressed her task, and using that code extensively throughout her `R` script without abstracting the data science concepts included in the code to broader classes of data science tasks.

Yet, other aspects of Alicia's code revealed an operational understanding of **data structures** in `R`, knowledge of the **`R` environment**, and how to create presentation-worthy **data visualizations**. Alicia used two methods to access data in `R`. When modeling, she exclusively used `$` to select columns from a `data.frame`, demonstrating an elementary understanding of the structure of a `data.frame`. When creating scatterplots, she exclusively used the `with()` function to temporarily attach the dataset, so she could name columns without selecting them with `$`. By not pairing `$` alongside the `with()` function Alicia demonstrated an understanding of how data are accessed by the `R` environment. However, when asked about the different methods of selecting variables from a `data.frame`, Alicia reiterated that she used whatever she had found online that worked, again suggesting she may have learned just enough to accomplish the specific task, without abstracting what she had learned in the process.

When creating the scatterplots associated with each simple linear regression, Alicia would typically declare axis labels, using the `xlab` and `ylab` arguments, and would always change the orientation of the y-axis labels, using the `las` argument. Alicia declared that she distinctly remembers learning how to modify plot axes in a lab for her GLAS I statistics course.

Spring 2019 Beginning the second year of her graduate program, Alicia had made modifications and additions to her previous `R` script, and had generated another `R` script for a different direction of her research. The data science skills seen in the fall persisted, with changes primarily to the themes of **data wrangling**, **workflow**, and the **`R` environment**, as seen in Figure 4.6.

In Alicia's modified code, she changed from sporadically creating code comments to

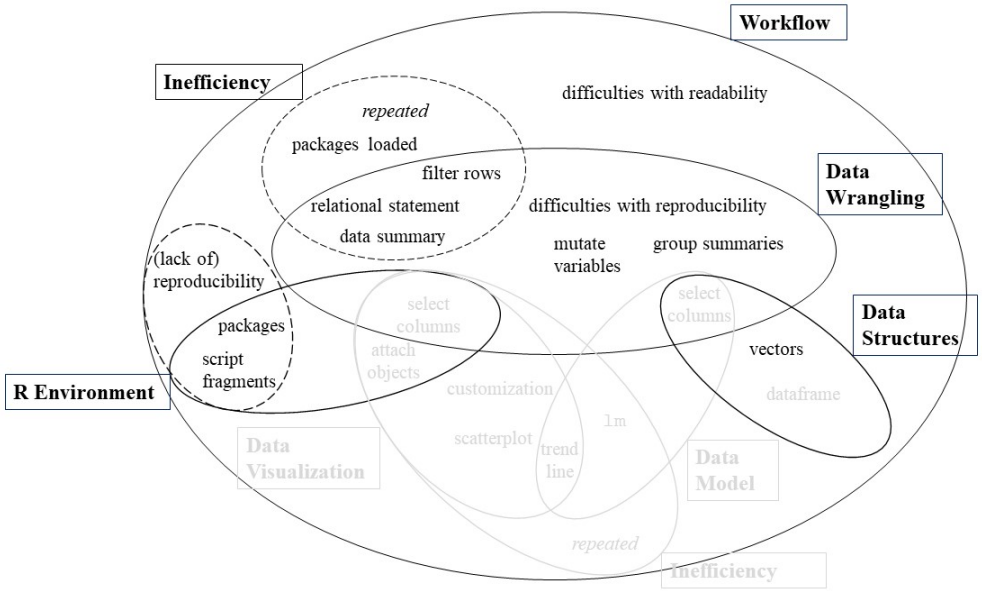


Figure 4.6: Concept map of data science skills seen in Alicia's `R` scripts in the spring of 2019. Previously existing data science themes that saw new skills appear continue to be outlined in black; new data science themes and skills appear as ellipses with dashed borders; themes and skills seen previously that experienced no changes appear in grey.

a more consistent workflow, creating a code comment prior to each data investigation. Additionally, Alicia created heading and subheading comments describing the overall purpose of the `R` script and the purpose of each section of code. Although Alicia continued to load her data in through the RStudio GUI, in her revised code, Alicia made use of the built-in `log()` function to make scripted mutations of existing variables, rather than making these mutations in Excel. However, in this `R` script, Alicia continued to use Excel to filter her data.

Unfortunately, with these additions to Alicia's `R` script came hiccups in her workflow. In this modified code, non-existent objects were referenced on occasion. These script fragments, referencing non-existent objects, all stemmed from a mistake in the capitalization of the name of an existing object. When questioned about these code statements, it took Alicia opening her `R` script and running the code to believe that these statements

would return an error. These script fragments, thus, led to the emergence of skills at the intersection of **workflow** and the `R` **environment**, a new manifestation of Alicia's struggles with a reproducible workflow.

The new `R` script Alicia generated served a different purpose than her previous `R` script. The purpose of this new script was to generate summaries for different subsets of her data. Hence, the skills within the theme of **data wrangling** seen in this script differ substantially from what had been seen previously. Alicia's previous use of `$` to select columns from a `data.frame` was again used throughout her new `R` script. However, in this new script, rather than relying on Excel, Alicia used reproducible `R` scripts to filter her data. Due to the substantial differences in Alicia's methods for working with data between her two `R` scripts, a transition to "difficulties with reproducibilty" was seen in within the data science skills associated with the theme of **data wrangling**.

In this new `R` script, Alicia used brackets (`[]`) to filter elements from the column which had been selected. This pairing of `$` with brackets demonstrates a deeper level of understanding of **data structures** than had been seen in Alicia's previous scripts. Often, the elements of these columns were filtered using an equality relation (`==`), testing whether the value of a specific column was equal to a specified discrete value, as seen in the statement below.

```
RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 1]
```

Unfortunately, from this data filtering process emerged a new instance of the theme of **inefficiency**. Alicia repeated this process 18 times, modifying only the discrete value checked against the "Age" variable. This data filtering method differs, however, from the other method Alicia used to filter data in this `R` script, namely her use of the `subset()` function. Alicia used the `subset()` function to filter data based on ranges of quantitative variables, and to join consecutive filters, as seen in the statement below.

```
mid <- subset(RPMA2Growth, StockYear < 2014 & StockYear > 2003)
```

In these cases, Alicia always used `>` or `<` relational statements, which were often joined with the logical value `&`. As seen in the modifications Alicia made to her original `R` script, in this new `R` script Alicia continued to use scripted variable mutations. However, these mutations primarily took the form of changing the data type of an existing variable using the `transform()` function.

Finally, the previous data summaries seen in Alicia's code consisted of using `mean()` and `sd()` to find summaries of specific variables. In this `R` script, however, Alicia transitioned to finding the mean of a variable across groups of another variable. Alicia executed this grouped data summary in two ways: (1) by making use of the `ddply` function, or (2) by operating the `mean()` function on a filtered dataset, using the bracketed method described above. These group summaries were then used to either create visualizations of the group means, or to create a new `data.frame` of these data summaries. When assembling the data summaries into a new `data.frame`, however, it became clear that Alicia did not have the foundational understanding of vectors necessary to construct a `data.frame`. As seen in the statement below, Alicia attempted to input all of the grouped summary objects individually where a vector of inputs belongs. This misunderstanding caused us to wonder if the syntax for the sequence of numbers input for `Age` led Alicia astray, as there is no outward indication that `R` considers this sequence of numbers to be a vector rather than a sequence of individual elements.

```
x <- data.frame("Age" = 1:9, "Growth" = Weight1, Weight2, Weight3,
                                        Weight4, Weight5, Weight6,
                                        Weight7, Weight8, Weight9)
```

Although this new `R` script contained many new data science skills, Alicia's extensive use of code comments in her modified `R` script was not seen in this new script. Similar to Alicia's transition to "difficulties with reproducibility," because of these contradictions in the readability of Alicia's scripts, a transition to "difficulties with readability" was also seen. The sole comment retained in Alicia's new `R` script, "`#Tanner's code/help`," suggested that portions of the script were not generated by Alicia. This aligns with the code statements

repeated verbatim throughout the script, and the redundant and haphazard loading of `R` packages. When questioned about how Tanner had helped her with her code, Alicia stated that he had given her code to "get group means" and to "subset her data" based on the value of a variable. Alicia stated that she tried the code Tanner gave her to subset, the method which used brackets and an equality relation, and it worked. So, she "copied and pasted the code for every subset [she] wanted." This led us to believe that Alicia was unaware of the differences in the two methods used in her code to filter data, and, likely, was unaware of the nuances of the data structures being employed by `R` when filtering variables using brackets and an equality relation.

Fall 2019 At the final data collection, Alicia had just proposed her research to her committee. For her presentation, she had made substantial modifications to her first `R` script, from the fall of 2018. While Alicia made modifications to the `R` script she had generated in the spring of 2019, these modifications were entirely organizational, with no new lines of code produced. Thus, only the third iteration of the `R` script from the fall of 2018 was analyzed.

In the fall of 2019, this `R` script had grown substantially, with a total of 870 lines, a dramatic increase from the 182 lines seen in the spring of 2019. The themes of data science skills seen previously continued in this modified `R` script. However, the largest changes were seen in the themes of **workflow**, **data visualization**, and **data wrangling**, demonstrated by Figure 4.7.

As one might expect, an `R` script with over 870 lines can be difficult to organize in a cohesive manner. Although Alicia retained her previous practice of using code headings and subheadings and comments before each data model, her pervasive use of the same model fitting, model visualization process made her code difficult to organize. Solely using code comments with 97 different data models makes it nearly impossible to organize a model selection process. Artifacts of the need to create this organization were seen in the revisions Alicia made to the content of the code comments before each

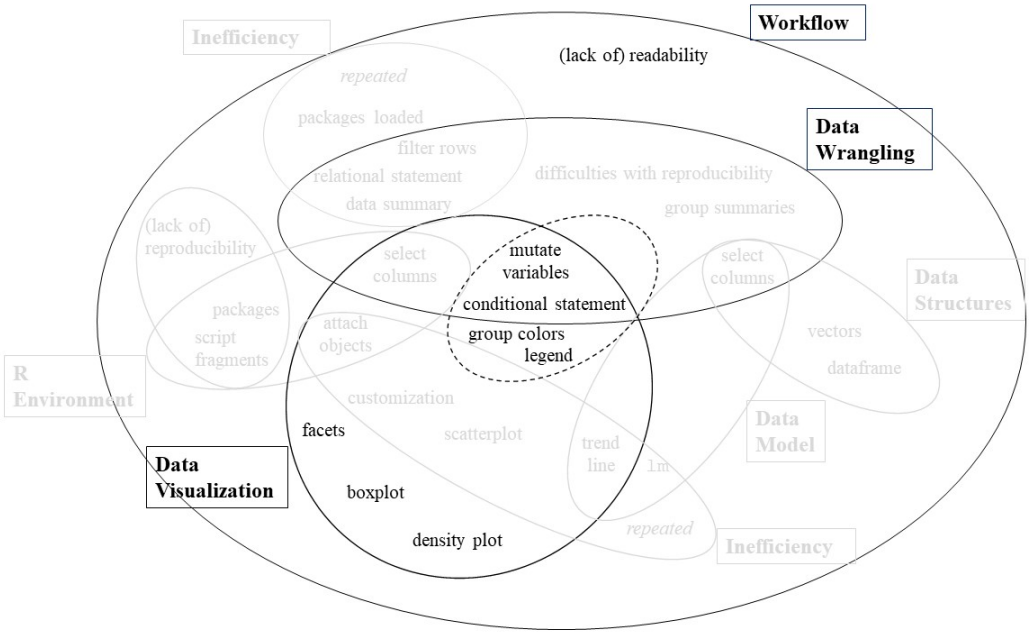


Figure 4.7: Concept map of data science skills seen in Alicia's `R` scripts in the fall of 2019. Previously existing data science themes that saw new skills appear continue to be outlined in black; new data science themes and skills appear as ellipses with dashed borders; themes and skills seen previously that experienced no changes appear in grey.

model. Comments gained descriptions of the model's fit, such as "`#Avg of AM -- good`" or "`#Favorite Regression`," or descriptions of some aspect of the model's fit directly after the model summary, such as "`summary(lmKnLipidAllMidKn) #.2389`." Furthermore, fitting 97 different data models resulted in a large number of additional organizational difficulties. First, the creation of 97 different intermediate objects left little room for each object to possess an intuitive name for what it contained. Second, the `R` script became a graveyard of code executed to recall the attributes of a given model or code that Alicia had attempted to run, but found an error. Lastly, all of the references to non-existent variables remained; however, unlike Alicia's previous `R` script, in this modification, these references were commented out. Each of these difficulties contributed to Alicia's **workflow** reverting to a "(lack of) readability."

All of these organizational difficulties aside, this final version of Alicia's code had

an organization of package installation and usage not seen in her previous versions. In Alicia's previous `R` scripts, packages were loaded haphazardly, often after the scripts using them or redundantly, both at the beginning and the end of the `R` script. Instead, in this version of Alicia's `R` script, the packages used previously were installed and loaded at the start of the `R` script. Then, when functions from the packages were being used, a code comment proceeded the statement, making it clear that the `lattice` (Sarkar, 2008) or `plyr` (Wickham, 2011) package was being used. The exception, however, comes at the end of Alicia's `R` script, where she tackled the problem of creating an interaction plot for a data model by using a function from an `R` package available on GitHub. Although the placement of these statements contradict her previous organization, the implementation of the function has a sense of organization. Prior to the installation of both the `devtools` (Wickham et al., 2019) and `ggiraphExtra` (Keon-Woong, 2019) packages, Alicia added comments describing why each package was necessary. Next, she copied-and-pasted an example of how to use the function she wished to use from the `ggiraphExtra` (Keon-Woong, 2019) package. Finally, she used the function to create the plot, accompanied by a comment describing the type of plot the function produces.

The inclusion of multiple predictors in her linear regression warranted a change in the **data visualization** tools Alicia had used, with a need to produce multivariate data visualizations. Alicia addressed this formidable task in the way many new `R` users might—by applying code that worked for a bivariate relationship to a multivariate relationship. Specifically, Alicia attempted to modify the process she had repeated over and over before, by fitting a multiple linear regression model, then producing a multivariate data visualization that would incorporate the trend line captured by the data model. An example of this process is shown below.

```
multAllE <- lm(PADataNoOutlierMultMeasure$Energy ~
                log(PADataNoOutlierMultMeasure$PSUM) +
                log(PADataNoOutlierMultMeasure$PSUA) +
                log(PADataNoOutlierMultMeasure$PSUP))
```

```
plot(PADataNoOutlierMultMeasure$Energy ~
     log(PADataNoOutlierMultMeasure$PSUM) +
     log(PADataNoOutlierMultMeasure$PSUA) +
     log(PADataNoOutlierMultMeasure$PSUP))

abline(multAllE)
```

While this process was not fruitful, Alicia was able to incorporate additional information into the bivariate plots she had produced previously, through the use of additional `R` packages and color. As eluded to with her use of a GitHub package in her revised `R` script, Alicia used the `ggiraphExtra` (Keon-Woong, 2019) package to create visualizations of an interaction model, as seen in the statements below.

```
# Fit the interaction model you are interested in

BigEnergylmLengthWeightInt <- lm(largeNoPrelim$Energy ~
            log(largeNoPrelim$PSUM) +
            log(largeNoPrelim$`Fork Length`) +
            log(largeNoPrelim$Mass) +
            log(largeNoPrelim$`Fork Length`)*log(largeNoPrelim$Mass))


# Plots the interaction with different colored points
and lines for the weights

ggPredict(BigEnergylmLengthWeightInt, interactive = TRUE)
```

Alicia also used the `lattice` package to infuse facets into univariate density plots and bivariate boxplots. Facets allow for visualizations to be partitioned into a matrix of panels, where each panel displays the relationship in question for a different subset of the data. Alicia used facets to split density plots and boxplots into groups of a quantitative variable. Lastly, plots Alicia had produced earlier, using the familiar `plot()` function, now included colors for different groups. To color points in a scatterplot, Alicia used the `ifelse()` function to obtain the colors, by mutating an existing variable. For these scatterplots, the conditional statement compared a variable not included in the plot to a hard-coded number,

declaring colors for points that satisfied and did not satisfy the relational statement, as seen in the statement below.

```
with(PADataNoOutlier, plot(Lipid ~ log(PSUA), las = 1,
     col = ifelse(PADataNoOutlier$`Fork Length` <  260,
                  "red", "black")))
```

This new process of coloring points, however, surfaces connections with themes of data science skills seen previously. The action taken by this statement is connected to the themes of **data wangling** and the **R environment**. In this statement, it is not difficult to see that Alicia is mutating an existing variable, as she had done previously; however, the intricate connection with Alicia's understanding of the R environment may be less obvious. The `with()` function is retained, as that was the method Alicia had used to create every scatterplot. The `$` was added to the statement only when the conditional statement was added, perturbing her data visualization system. The combination of these two approaches, however, makes it clear that Alicia was unable to abstract her understanding of how R accesses data beyond her working solutions. This lack of understanding is reiterated when, at a later point in her code, Alicia chooses to `attach()` her data, but then continues to access columns from the data using a `$`.

Alicia's Data Science Trajectory Over the course of the study, Alicia's data science skills experienced a great deal of change. In the fall of 2018, the R script Alicia had produced contained 72 lines of code, with the process of fitting a model, visualizing the model, and inspecting the model, repeated six times. Alicia exclusively used Excel to manipulate her data and her R script exhibited little organizational structure, due in part to a lack of code comments. In the spring of 2019, Alicia's two R scripts possessed contradicting data science skills. In her revised R script, Alicia made extensive use of code comments, to describe each of the data models she was fitting and to delineate sections of the R script. But in her new R script, few, if any, code comments were present. Additionally, while Alicia continued to use Excel for her data manipulations for her revised R script, in her new script, Alicia used

`R` to wrangle her data. Interestingly, in Alicia's new `R` script, there were signs of assistance she had received from a peer. Unfortunately, both `R` scripts contained statements of code that would not run, due to packages not being installed or the incorrect capitalization of variable names, adding a new instance of unreproducibility in Alicia's code. In the fall of 2019, disappointingly, Alicia's `R` script which previously had extensive comments grew to such a size that code comments became unwieldy. While the readability of this modified `R` script was lacking, Alicia did employ new data science skills to create multivariate data visualizations. As opposed to her previous `R` script, which suggested Alicia had received help from a peer, to make these multivariate plots, Alicia used resources from GitHub. The "tutorial" Alicia left herself surrounding these GitHub resources suggests the possibility that Alicia was moving toward being a more independent learner.

Although Alicia used new data science skills during the study, there is little evidence that her mental models of the `R` environment grew. In the fall of 2018, Alicia used a `$` to select columns from a dataset when fitting a model, and used `with()` to attach a dataset when producing a visualization. Each of these skills remained unchanged throughout the duration of the study, with no sign that Alicia was able to abstract the concepts included in these tasks to broader classes of data science tasks. Thus, when acquiring additional data science skills, Alicia used these broken mental models. In the fall of 2019, to create group colors in her scatterplot, Alicia made use of the `ifelse()` function. The way in which Alicia interacted with this function, however, reiterates her inability to abstract the data science skills she was familiar with.

Ellie

The spring of 2018 was Ellie's second semester as a graduate student, pursuing an interdisciplinary Master's in Ecology and Environmental Sciences, en route to a doctorate. Ellie had completed a Bachelor's degree in engineering at a medium-sized research university in the Midwestern United States. During her undergraduate degree, Ellie's coursework entailed extensive programming experiences in MATLAB and multiple courses in GIS.

During her first semester as a graduate student, Ellie completed the "R programming" module in `swirl` (Kross et al., 2018), at the recommendation of her adviser, due to the extensive amount of computational work done in their lab. In her second semester, Ellie enrolled in GLAS I, also at the recommendation of her adviser. Within the duration of this study, Ellie also took GLAS II in the fall of 2019.

In addition to her statistics courses, Ellie completed a variety of quantitative methods courses specific to environmental sciences. Each of these courses used `R` exclusively, targeting a variety of different computing concepts. One course focused on how to manage environmental data, with topics ranging from understanding the inner workings of `R`, to importing, tidying, and reorganizing data for statistical analysis, to integrating `R` with databases. The other course focused instead on measurement methodologies in environmental sciences, with topics covering a broad array of measurement devices, as well as programmatic approaches to measurement and uncertainty analysis. Potentially due to her coursework preparation and undergraduate experiences in MATLAB, Ellie did not attend any university-sponsored or external `R` workshops. As is typical for graduate students in this field, at this university, the direction of Ellie's program was largely dependent on her adviser. With a strong background in computing and research that focused on pairing data with simulation modeling to understand environmental systems, Ellie's adviser understood the importance of her computational and statistical preparation.

Over the duration of the study, Ellie produced multiple `R` script files for her research. These scripts fit largely into two categories: `R` scripts generated for the production of `R` packages, and `R` scripts generated for research presentations.

Fall 2018 In the fall of 2018, Ellie was producing scripts for a variety of purposes: for her own research, for work she was doing in an environmental monitoring course, and to make contributions to an `R` package in development by members of her lab. The name of the `R` package to which Ellie made contributions is omitted for anonymity. To make contributions to this `R` package, Ellie pushed her changes to the package's GitHub repository,

using the RStudio interface. But Ellie's use of version control captures only a small glimpse into her efficient, reproducible, and readable **workflow**. The full set of data science themes seen in Ellie's code are outlined in Figure 4.8.

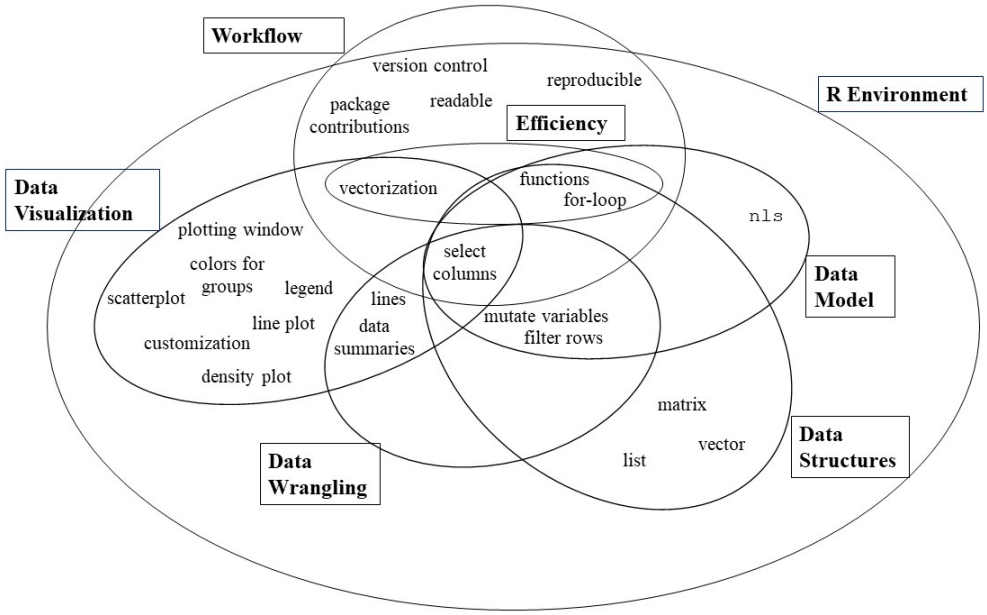


Figure 4.8: Concept map of data science skills seen in Ellie's `R` script in the fall of 2018. Themes are loosely arranged into groups, with theme names described in boxes. The list of specific skills seen within each theme are included in the ellipses. The skills included are not exhaustive, but rather are indicative of the type of skill seen in the research code. Many skills fall into multiple themes, as evidenced by the overlapping ellipses and the appearance of the same skill in multiple locations.

When inspecting Ellie's `R` script, one immediately notices the attention to detail Ellie put forth in making her `R` scripts as readable, reproducible, and efficient as possible. In her `R` scripts, Ellie first sourced in functions she had written previously, then loaded in all of the packages needed, and finally imported the data necessary for her analyses. Throughout the `R` script, Ellie made extensive use of code comments, both to describe the actions being taken in the code and for clear separation of sections of code. When questioned about the nature and contents of the `R` scripts containing functions that were sourced in, Ellie stated that they were `R` scripts for functions she had used throughout her code for the course, as

well as functions developed by Ellie and her labmates for their `R` package. When asked about where she learned to write functions in `R`, Ellie said that during her undergraduate degree she "got some practice writing functions and for-loops," and through taking the `swirl` course she got the refresher she needed to remember "how to code and how to write functions." Ellie's fluency with writing her own `R` functions foreshadows her understanding of both the **`R` environment** and the **efficiency** of her coding practices.

Throughout Ellie's research code, she iterated through the process of creating estimates using different standard deviations and plotting the results. For three values of standard deviation, Ellie implemented a for-loop to iterate through realizations, adding to her knowledge of methods to accomplish repeated processes. Prior to each for-loop, Ellie initialized empty vectors and a matrix, specifying the dimensions of each object using previously defined variables. During each for-loop, Ellie employed data science skills related to **data wrangling** and **data structures**. At each iteration, Ellie filtered rows or selected elements from these objects using brackets, which included the value of the iteration index. The values saved in these objects were then mutated to calculate and store values of other objects. Notably, during this process of data wrangling, Ellie paid careful attention to not repeat the same calculation twice. Note in the statement below, estimates from the `predict()` function are only calculated once, and the resulting value is used for each subsequent calculation.

```
# Store model predictions for the estimated parameters
confidences1[realization,] <- predict(nlsr)

# Calculate the standard deviation of the residuals
sderr <- sd(synthData - confidences1[realization,])

# Store model predictions with added error
# to generate prediction intervals
predictions1[realization,] <- confidences1[realization,] +
  rnorm(length(time), mean = 0, sd = sderr)
```

In the final portion of Ellie's code, she created both univariate and bivariate visualizations of her model simulations. The primary **data visualizations** Ellie created

were density plots of parameter estimates, and scatterplots to visualize how parameter estimates varied over each iteration. Every plot Ellie created contained customizations for axis labels, orientation of y-axis labels, and x- and y-axis limits. The density plots contained information about two or more groups, necessitating the inclusion of colors. For each group represented in the plot, Ellie plotted its density, followed by vertical lines reflecting 95% confidence intervals for the group's estimates. Because of the colors included in the plot, each plot contained a legend, noting what color corresponded to which group, and a label for each group.

Ellie's understanding of data structures and the `R` environment was also seen in her inclusion of the vertical lines in the density plots of each parameter. The location of these lines was defined by selecting an element from a vector of data summaries, using brackets and a column's name, as seen in the statements below.

```
confidenceResp3 <- quantile(respRateEst3, probs = c(0.025, 0.975))
abline(v = confidenceResp3["2.5%"], lty = "dashed",
       col = "firebrick1")
```

Ellie's calculation of the confidence and prediction intervals for each scatterplot, however, required a different approach. To accomplish this task, Ellie made use of the `apply()` function, calculating a vector of quantiles for each column of a matrix. Ellie's use of functions that allow for vectorized calculations further adds to her knowledge of tools available in `R` for repeated operations. As seen in the statement below, it should be noted that throughout Ellie's `R` script, in every instance where she uses a function, regardless of the function's origin, Ellie named each argument to the function. Stemming perhaps from her experiences creating user-defined functions in `R`, this coding practice reflected both Ellie's understanding of the `R` environment, and her appreciation for a readable data workflow.

```
confidenceInt3 <- apply(
   X = confidences3,
   MARGIN = 2,
   quantile,
   probs=c(0.025, 0.5, 0.975)
)
```

Spring 2019 In the spring of 2019, Ellie began work as a research fellow on a grant building stream metabolism models. Ellie's work was under a member of her doctoral committee, who had a background in software design. Due to her committee member's background, Ellie was encouraged to learn and create an `R` package using R6 classes. `R6` is a package in `R` which allows for the "implementation of encapsulated object-oriented programming," (Chang, 2019). Because of its object-oriented nature, `R6` is a "simpler, faster, lighter-weight alternative" to create objects (classes), rather than using `R`'s built-in classes. Central to object-oriented systems are the concepts of class and method. Object-oriented systems "call the type of an object its **class**, and an implementation for a specific class is called a **method**" (Wickham, 2019, emphasis in original). An object's class defines "the data possessed by every instance of that class" (Wickham, 2019), called fields. Base `R` has three object-oriented systems, S3, S4, and Reference classes (RC), each differing in how objects and classes are defined. R6 is provided by the `R6` package, and has similarities to RC, but uses S3 rather than S4.

Below is an R6 class among the classes defined in Ellie's code. With R6, the `R6Class()` function is the only function used to create both classes and methods. The first argument, `classname`, is the name of the class, and the second argument, `public`, is a list of methods and fields. Methods (functions) can access the methods and fields of the object using `self$`. A new object can be created from a class by calling the `new()` method. Including an `initialize` method in the class definition "overrides the default behavior of `new()`" (Wickham, 2019).

```
# Load packages
library(R6)

#### State ####
State <- R6Class(
  classname = "State",
  public = list(
    name = NULL,
    value = NULL,
    is.dynamic = NULL,
```

```
    initialize = function(name = NULL,
                          is.dynamic = F,
                          initVal = NULL){
      self$name <- name
      self$is.dynamic <- is.dynamic
      self$value <- initVal
    },
    calculate = function(){
      stop("calculate method does not exist for this class")
    }
  )
)
```

Ellie stated that, prior to her work in her lab, she "didn't even know what object-oriented programming was," but through her extensive computing experiences in her lab, she was asked to participate in this committee member's research. Potentially because of the committee member's background in software design, they held a preference for working with the `R6` system, as for individuals familiar with object-oriented programming, `R6` "will feel very natural" (Wickham, 2019).

The `R` scripts Ellie submitted reflected the initial process she had made toward learning and applying this new system. Thus far, Ellie had produced four `R` scripts containing functions for three portions of the `R` package, one `R` script defining the R6 classes, and one "sandbox" `R` script implementing the model. Not unlike learning a new programming language, Ellie's knowledge of the **`R` environment**, **data wrangling**, **workflow**, and **efficiency** carried over into this new system. However, with this new system, Ellie exhibited new data science skills in each of these themes, as seen in Figure 4.9.

In Ellie's definitions of the R6 classes for the model, she continued to make use of user-defined functions and vectorization for repeated processes, and continued to use bracketing to select variables. This new system, however, brought the need for Ellie to expand the data science tools she had been using previously. Broadening her use of the `apply` family of functions and demonstrating her ability to flexibly work with different data structures, Ellie made use of the `lapply()`, `sapply()`, and `mapply()` functions. To

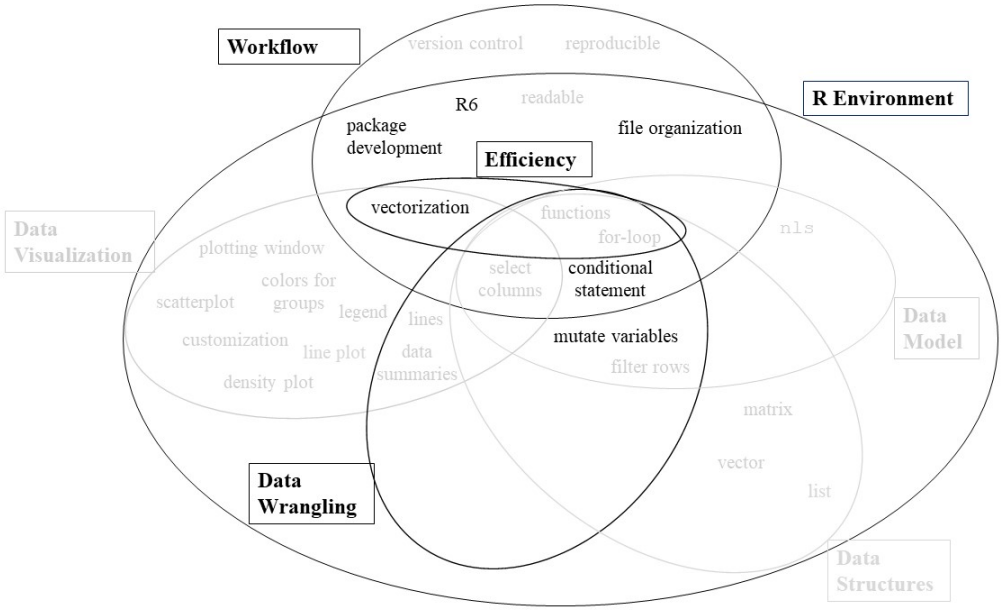


Figure 4.9: Concept map of data science skills seen in Ellie's R scripts in the spring of 2019. Previously existing data science themes that saw new skills appear continue to be outlined in black; new data science themes appear as ellipses with dashed borders; themes and skills seen previously that experienced no changes appear in grey.

implement function breaks, Ellie utilized conditional (`if()`) and relational statements to test if an input failed to satisfy a condition, as seen in the statement below.

```
if(length(x) >= 1){
        stop(print("The Delta", self$name,
        "has a processName that matches multiple processes
        in the model"))
      }
```

Lastly, developing an R package of her own, Ellie made modifications to her workflow. Ellie used an RStudio project to keep all of the files associated with the R package together. This file storage system allowed for Ellie to sync her materials to the project's collaborative GitHub repository, where Ellie continued to use the built-in GitHub interface within RStudio.

Fall 2019 Over the summer of 2019, Ellie continued to work on two R packages—one associated with the work done by her lab, and one she had developed the previous spring. In preparation for a committee meeting, Ellie had generated an R script implementing a maximum likelihood analysis of her experimental data. As opposed to her previous R scripts, this script focused on implementing a specific data model to Ellie's own research data, rather than creating an R package to produce data models for a variety of data. Potentially due to the applied nature of the script, a large number of new data science skills appeared inside the existing theme of **data wrangling**, alongside changes to the themes of **data structures** and **workflow**. These new skills continued to map the interconnected nature of the "key" data science skills Ellie employed throughout her research, as seen in Figure 4.10.

The R script Ellie generated, did not have the same **workflow** structure as her

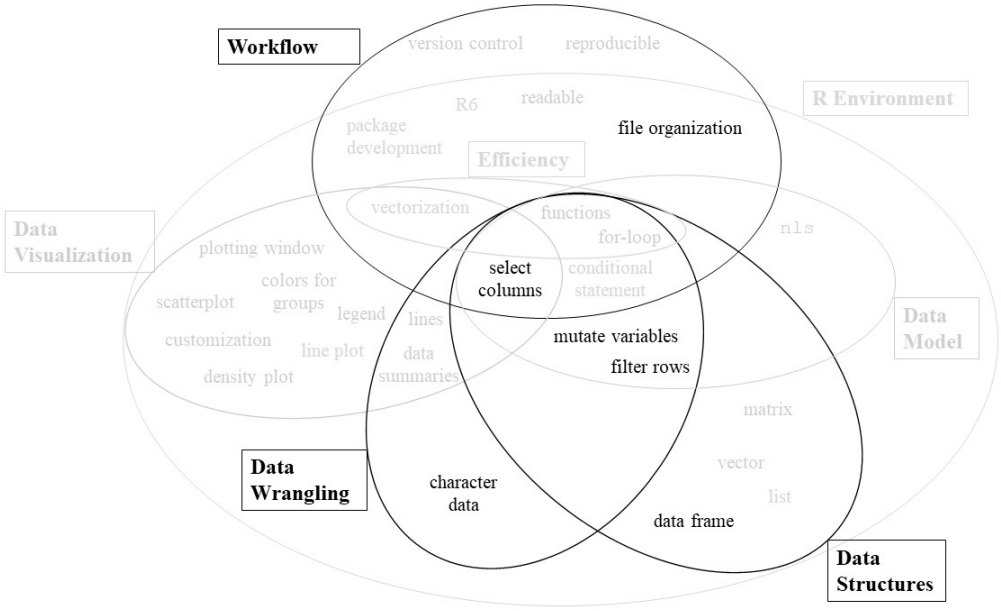


Figure 4.10: Concept map of data science skills seen in Ellie's R scripts in the fall of 2019. Previously existing data science themes that saw new skills appear continue to be outlined in black; new data science themes appear as ellipses with dashed borders; themes and skills seen previously that experienced no changes appear in grey.

previous research projects. This project had a different file structure, where the data were not stored in the same folder as the `R` script, requiring Ellie to specify a full path to access the data, rather than the relative paths used previously. Potentially due to the use of real data, Ellie's script made use of a `data.frame` instead of a matrix. Because of this change in **data structures**, Ellie used new **data wrangling** techniques to filter rows, select columns, and mutate variables. Previously, Ellie had exclusively selected columns from a matrix or vector using brackets, with a character vector of names. Throughout this `R` script, she used both `$` and hard-coded numbers to select columns. Where Ellie had previously only used brackets with indices to filter rows, in this `R` script, she additionally used the `subset()` function and brackets with logical values. Finally, Ellie utilized a variety of new methods when mutating existing variables, such as creating substrings, using the inclusion operator (`%in%`) in a relational statement, changing data types, or using hard-coded calculations. Examples of these techniques are highlighted in the statements below. The second statement highlights Ellie's ability to filter data in a variety of ways.

```
gas <- gas[!(substr(gas$sampleID,3,3) %in% c("b","c")), ]

N15_NO3_O_D <- 40*((carboys[carboys$CarboyID == "D",]$EstN15NO3) +
     (0.7*RstN/(1+RstN)))/(subset(gas, gas$carboy == "D")$Ar[1])
```

As evidenced by the statements above, Ellie's `R` script continued to highlight her fluency in data structures. Throughout her code, Ellie displayed an understanding of the relationships between different data structures and the flexibility to modify her data wrangling techniques to suit the data structure with which she was working. Ellie readily selected columns from a filtered `data.frame`, or selected indices from the resulting vectors. In the statement below, Ellie selected a column and associated rows from a `data.frame`, then filtered an element of the resulting column, both using brackets. Reiterating Ellie's attention to duplication, the variables included in each step were defined previously.

```
lowerCIBound <- pMat[1:mlleIndex,1][which.min(
                        abs(
```

```
mlleCI+likelihoods[1:mlleIndex]
)
)]
```

Ellie's Data Science Trajectory The ease with which Ellie transitioned her knowledge and skills in MATLAB to the `R` programming environment was seen throughout the study. After having completed the "`R` programming" module in `swirl` (Kross et al., 2018) the prior semester, in the fall of 2018, Ellie demonstrated both an extensive understanding of the importance of a data workflow and a commanding understanding of the `R` environment. Ellie's experiences with file organization, through her contributions to her lab's `R` package and through an environmental monitoring course, were evident in her use of separate `R` scripts for functions and analyses. Moreover, Ellie's `R` scripts were readable, with comments throughout, reproducible, using scripted data wrangling, and efficient, by not repeating itself. When questioned about where she learned a variety of data science skills, Ellie continually stated that she transferred her understanding of MATLAB into `R`. It is possible that once Ellie had an understanding of data structures and functions in `R`, she was able to begin expanding her skills to new concepts, such as vectorization. In the spring of 2019, Ellie continued to expand these skills, in the context of a new programming environment. Potentially because of her mastery of programming in `R`, Ellie took on a research position which required her to create an `R` package in the `R6` programming language. The project expanded her skills with data structures, vectorization, and function writing. Finally, in the fall of 2019, Ellie's `R` scripts demonstrated her commanding understanding of working in `R`. Ellie's ability to flexibly transition between various methods when working with data in `R`, speaks to her densely connected mental model (Wilson, 2019) of the `R` environment.

Cross-Case Discussion

Although similar themes of data science skills arose for both Alicia and Ellie, some aspects of these themes differed substantially, while others saw a large overlap. In this section, we dissect the similarities and differences of the data science skills used between these two cases. Specific attention is paid to each case's experiences in acquiring data science skills and the environments that fostered or inhibited learning.

The dramatic differences in the ease with which Alicia and Ellie worked in `R` can be seen in the differences in the structure of their concept maps. The **`R` environment** lies at the center of Ellie's concept map, as seen in Figures 4.8, 4.9, and 4.10, with substantial overlap between each of the seven themes exhibited in her code. Instead, **workflow** lies at the center of Alicia's concept map, as seen in Figures 4.5, 4.6, and 4.7, permeating the remainder of the themes seen in Alicia's code. Where Ellie's map shows a large degree of overlap, considerable disconnections exist between each of the seven themes seen throughout Alicia's code. These connections speak to the level of fluency for working in `R` with which both women possessed. The densely-connected network of data science skills in Ellie's concept map communicates her fluency for working in `R` (Wilson, 2019). The disconnected nature of Alicia's concept map, however, contextualizes the brute force methods Alicia used to accomplish the tasks she faced. These two endpoints of `R` fluency help to illuminate hurdles students may experience as they build data science skills.

Whereas Alicia felt uncomfortable and unfamiliar importing her data using a script, Ellie used `R` scripts to source in her data and frequently sourced in `R` scripts containing functions she frequently used. Ellie's `R` scripts had a specific organization, beginning with a heading declaring the purpose of the script, then loading in the necessary data, functions, and packages. Each section of Ellie's `R` script could be easily followed, as she used the same heading format for each section. Furthermore, within each section, Ellie made intentional use of comments, describing the action each statement of code was executing. Meanwhile, Alicia's code was sparsely organized, with code comments appearing for only a glimpse of

time. Alicia's R scripts tended to look like a "code graveyard," containing every statement of code she generated for a specific task, including the fragmented statements that would not execute. This code graveyard also contained the haphazard loading of R packages, often after functions from the package had been used. As Alicia's script became longer and longer, she began to comment out statements of code, with no indications why the specific statements were no longer needed. Alternatively, the few statements commented out in Ellie's code were associated with a comment why she believed the statement may not be necessary.

As stated by Alicia, to accomplish many tasks, she used whatever she had found online that worked. This inefficient, brute force approach was seen in Alicia's extensive use of the copy-paste-modify approach to accomplish tasks that required repeating a large number of times, whereas, Ellie's code adhered to a central tenant in good coding practice—"be ruthless about eliminating duplication" (Wilson et al., 2017, p. 3). Instead of the copy-paste-modify paradigm, Ellie utilized for-loops, user-defined functions, and vectorization to eliminate repetition in her code. This consequential difference in Alicia and Ellie's ability to "decompose programs into functions" and eliminate repetition speaks to the substantial differences in their mental models for working in R.

These large differences in levels of R fluency continued to be seen in the substantially different ways with which Alicia and Ellie worked in the R environment. On the one hand, Alicia experienced hurdles understanding the way R accesses data, frequently using `with()` and `attach()` to attach a `data.frame` to the R search path, while redundantly continuing to select columns from the `data.frame` using a `$`. This misalignment of ideas is not surprising when one remembers that when approaching a task in R with which she was not familiar, it was standard for Alicia to find an answer that worked on the internet. These piece-by-piece solutions never allowed for Alicia to "abstract what she learned from each task to broader classes of tasks" (Nolan and Temple Lang, 2010, p. 100). On the other hand, with a firm understanding of the R environment in hand, Ellie was able to use her understanding to

write more efficient code, frequently juggling both globally and locally defined functions. Furthermore, Ellie's fluency in `R` opened the door for her to contribute to a research project developing an `R` package using `R6` classes.

Potentially due to Alicia's difficulties navigating the `R` environment, she exclusively worked with data frames throughout her `R` scripts, with only a single instance of a vector in her code, and no instances of lists. Yet, after working with a `data.frame`, filtering rows to calculate a data summary, Alicia attempted to assemble a vector of data summaries into a `data.frame`. Alicia's mistake in creating this summary `data.frame` demonstrated her fragmented understanding of data frames. In contrast, Ellie's understanding of data structures allowed her to flexibly move from vectors to `data.frame`s to lists. The differences in these women's understandings of data structures became more pronounced in the data science skills related to wrangling data.

It should be emphasized again that, between the fall of 2018 and spring of 2019, Alicia transitioned from using non-reproducible methods to manipulate her data, to using scripted data wrangling. Specifically, Alicia transitioned from manually manipulating variables in Excel to mutating variables with base `R` functions, and from manually filtering data in Excel to filtering data in `R` using brackets or the `subset()` function. Elements of Alicia's transition reflect the hurdles scientists in a similar position may face. The data science techniques Alicia used to wrangle her data depended entirely on the task at hand. If the variable by which she wished to filter was quantitative, the `subset()` or `which()` functions would be used. If the variable by which she wished to filter was categorical, a relational statement would instead be inserted into brackets. Alternatively, Ellie showed the flexibility to transition between different data wrangling techniques in the same statement of code. She could filter data using a built-in `R` function, indices, or logical values, and she could select data using a `$`, brackets, or a character vector of column names. Ellie's mental model of data structures allowed for her to "switch back and forth between different views" of a data structure, to better suit the task at hand (Petre and van der Hoek, 2016).

Although Ellie and Alicia showed substantial differences in their knowledge and skills when working in `R`, the visualizations they each created used nearly identical skills. Both women created predominantly bivariate visualizations using scatterplots, and univariate visualizations using density plots. These data visualization procedures perhaps stem from their experiences in the GLAS I course, where these were the primary visualizations for univariate and bivariate relationships. Alicia and Ellie both added colors for groups to their visualizations, either by using a conditional statement or by specifying the color of additional points and lines. When adding these colors to their plot, both women created legends, differing only in their placement. Furthermore, both women added additional lines to their plots, both using `abline()`. Throughout the `R` scripts, Ellie would add vertical lines to her plots, corresponding to quantiles of a vector, and Alicia would add linear trend lines, corresponding to the variables included in each scatterplot. When Alicia needed to include four plots in the plotting window, she specified the graphical parameters using a vector of rows and columns (`par(mfrow = c(2,2))`). When Ellie faced this same task, potentially because of her thorough understanding of the `R` environment, she instead specified graphical parameters by directly modifying the plotting margins, using the `mar`, `oma`, and `xpd` parameters. Unlike Ellie, Alicia, however, attempted to infuse additional variables into her plots through the use of interaction plots and faceting. Notably, for every plot Alicia and Ellie created, they employed the same plotting customizations, consistently changing both the axis titles and the orientation of axis labels. When both women were questioned regarding these specific customizations, they emphasized that they had learned these specifications in a data visualization "best practices" lab in GLAS I. While Ellie's data visualization skills continued to expand throughout her independent research, the majority of the data visualization skills seen in Alicia's `R` code were acquired solely through one lab in her GLAS I course. The variation in the impact of the GLAS I course on the learning trajectory of Alicia and Ellie potentially arises from the differences in the support network of each woman during their independent research.

Where Ellie was able to transfer her understanding of MATLAB to `R` through a guided tutorial, Alicia began her graduate work with few prior programming experiences. After acquiring a foundational understanding of `R`, Ellie was able to expand her data science skills rapidly, whereas Alicia continued to haphazardly acquire data science skills through her use of the internet and her peer network. When questioned about the resources she used to acquire data science skills, Alicia continually lamented that she was provided little support from her adviser. Alternatively, from the outset, Ellie's adviser provided her with resources for acquiring critical data science skills, by recommending the `R` Programming course through `swirl` (Kross et al., 2018) and providing support for navigating difficult computational tasks. However, the support Ellie received from her adviser may not be the case for many environmental science graduate students, as dramatic changes to the computational landscape open the possibility that advisers may feel unprepared to support their students in acquiring the data science skills necessary for their research (Hampton et al., 2017; Nolan and Temple Lang, 2010; Teal et al., 2015). Although advisers are not expected to be able to provide computational support to their students, those unable to provide computational support themselves should be aware of internal and external resources that can support graduate students acquiring these necessary skills, as these resources could substantially impact a student's learning trajectory.

Implications

This research demonstrates the importance of data science skills throughout the entire data analysis cycle, and has implications for both statistics and environmental science educators. The importance of statistical preparation has been long understood by the environmental sciences; however, despite numerous researchers emphasizing the gravity of students' computational ill-preparation (Hampton et al., 2017; Hernandez et al., 2012; Teal et al., 2015; Strasser and Hampton, 2012), the field has yet to acknowledge the necessity of computing skills throughout the entire data analysis cycle. Alternatively, the

importance of data science training has been continually stressed by statistics educators (Baumer, 2015; Baumer et al., 2014, 2015; Cetinkaya-Rundel and Rundel, 2018; Cobb, 2015; Gould, 2010; Hardin et al., 2015; Horton and Hardin, 2015; Kaplan, 2018; Nolan and Temple Lang, 2010), but these conversations have notably overlooked a key population of statistics practitioners—graduate students from scientific fields.

This case study serves as an example, highlighting what data science skills may be necessary for students as they engage in data-intensive research, and the data science experiences that may be necessary for students to be prepared for independent research. While the spectrum of data science proficiency is vast, as evidenced by the dramatic differences between Alicia and Ellie's data science skills, educators have a duty to arm students with the foundational knowledge and skills that will shape how they perform their research. Every student should leave their scientific program with the ability to create a reproducible, readable, and efficient data workflow. Students need these workflows modeled for them throughout their curriculum, with tools such as `R` scripts, R Markdown files, RStudio projects, and version control. GLAS I, Alicia's sole quantitative preparation before performing research, provided no introduction to writing an `R` script to read in data, instead using data from the `Sleuth3` package (Ramsey et al., 2019), and no introduction to R Markdown files as reproducible documents that allow for written annotations. This lack of workflow preparation had pronounced impacts on the data science skills Alicia used throughout her research, as evidenced in her inability to import data without the RStudio GUI, and her attempts to create annotated summaries for her linear models with code comments. Thus, scientific educators need to realize the impact of the computing students see in their classroom on their future research.

Implications for Statistics Educators

Statistics educators should understand the necessity of data science skills for future statisticians and practitioners of statistics alike. As evidenced by the discouragement to take additional statistics courses by Alicia's adviser, faculty in environmental science fields

may assume that students are acquiring the data science skills necessary for the entire data analysis cycle in the single statistics course required for their degree. This assumption emphasizes the importance of incorporating data science skills across the entire statistics curriculum, including graduate-level service courses. With environmental science graduate students participating in independent research and analyzing their own data immediately after leaving the statistics classroom, the need for integrating data science into these classes cannot be understated. Using real data to introduce students to tools for reproducible data wrangling, reproducible statistical analyses should be modeled for students. Alicia's use of unscripted data manipulation, generation of `R` scripts riddled with duplication and without comments, and creation of projects devoid of organization, suggest that, until students see these practices modeled in the classroom, they may not know how to use them or grasp their importance.

Weekly data science labs are a powerful way to systematically provide students with these experiences throughout the course. Statistics educators should recognize the power of these labs on teaching students best practices for working with, modeling, and visualizing data. The GLAS I course taken by both Alicia and Ellie included a lab on "best practices" in data visualization. The skills Alicia and Ellie learned through this lab continued throughout their research code, consistently adding axis titles and changing the orientation of axis labels to improve each plot's readability. Throughout these labs, educators should model a set of "computing practices that every researcher can adopt, regardless of their current level of computational skill" (Wilson et al., 2017), so students understand their importance.

Finally, R Markdown documents and RStudio projects are invaluable tools for teaching best practices in data science. As Alicia's research progressed, her `R` scripts quickly became unmanageable, with no structure and disorganized comments regarding aspects of different models she wished to remember. Instead, an R Markdown document promotes an organized data workflow, through an "easy to use authoring framework for combining statistical computing and written analysis in one document" (Cetinkaya-Rundel and Rundel,

2018, p. 61). Furthermore, the rendered product that R Markdown creates discourages fragmented code lingering behind. Pairing R Markdown documents with RStudio projects models "good enough" practices for project organization. Throughout their research code, both Alicia and Ellie had a single statement of code at the beginning of each script which cleaned the file directory. By incorporating RStudio projects, students see firsthand what good project organization looks like.

Implications for Environmental Science Educators

As data science skills have grown to be necessary skills for research across scientific disciplines, research by environmental science educators has created a growing awareness of the computational ill-preparation of graduate students by their curriculum (Hampton et al., 2017; Hernandez et al., 2012; Teal et al., 2015; Strasser and Hampton, 2012). Thus far, this research has not acknowledged the interconnected relationship between these necessary data science skills and the implementation of statistical analyses. For students, "the challenges to data analysis start well before the computational steps involved with model fitting" (Nolan and Temple Lang, 2010, p. 99). Therefore, environmental science educators should recognize that the statistical analysis is only one component—and, often, a relatively small component—in the entire data analysis cycle. Emphasizing the data analysis cycle promotes persistent conversations surrounding best practices for scientific computing. Because it takes students time to abstract the data science skills they've learned to broader sets of tasks, it is not sufficient to expect that students are acquiring these skills solely through their statistics coursework. Instead, these concepts should be seen throughout the graduate environmental science curriculum.

Environmental science educators should recognize the importance of computing experiences for undergraduates as well. Ellie, with a firm understanding of MATLAB from her undergraduate degree, was able to easily pick-up how to code in `R`, by primarily using `swirl` to understand the syntax of a new programming language. Furthermore, once she had become fluent in the syntax of `R`, Ellie was able to rapidly progress her skills. Meanwhile,

Alicia, with little to no undergraduate programming experiences, relied entirely upon the data science skills she learned in her graduate-level statistics course, internet resources, and her peers. As Alicia attempted to transition from manipulating her data in Excel to reproducibly wrangling her data in `R`, Ellie advanced from writing for-loops to using `apply` statements in a matter of weeks. This raises the question: Why isn't every student provided with the opportunity to make these advances in their computing ability?

Limitations and Future Research

The research code produced by Alicia and Ellie serves as an example of the data science skills necessary for implementing statistics in data-intensive environmental science research. However, like any other data source, these artifacts have their advantages and limitations. One possible limitation lies in the information captured by these research artifacts. The research code produced by these participants may offer a fragmented view on the data science skills necessary for implementing statistics in data-intensive environmental science research. Some necessary skills may not be reflected in students' `R` code for a variety of reasons, such as students' limited knowledge of how to employ certain skills in `R` so they instead perform the operation in a different software, such as Excel. This possibility can be seen in Alicia's code, where data were manipulated in Excel, with only a single comment indicating how the datasets in the `R` script differed from each other. Despite this potential limitation, students' research code has a "stability" not present in interviews or observations (Merriam, 2009, p. 155), where students may fear researcher judgment for the methods they use to accomplish a task. As stated in the Introduction, artifacts of students' research are "objective" and "unobtrusive" sources of data (p. 155), as the presence of the researcher "does not alter what is being studied" (p. 155). Additionally, these artifacts are a "product of the context in which they were produced and therefore grounded in the real world" (p. 156).

Case study research offers a means of "investigating complex social units consisting of multiple variables of potential importance in understanding the phenomenon" (Merriam,

2009, p. 50), without "eliminating what cannot be discounted" or "simplifying what cannot be simplified" (p. 52). Though data science skills employed by Alicia and Ellie when implementing statistics in their research are anchored in the context of performing graduate-level data-intensive environmental science research, this research did not intend to assess the prevalence of these key data science skills across environmental science disciplines. The backgrounds and data science skills used by Alicia and Ellie may not represent the experiences of all environmental science graduate students, but much can be learned from the dramatic differences in how these women experienced this phenomenon.

This in-depth case study of the data science skills used by Alicia and Ellie throughout their research offers implications for future research on how students build understandings of data science concepts and how familiarity with other programming languages transfers to students' abilities to work with data. The difficulties Alicia experienced connecting data science concepts suggest the need for future investigations of how students learn data science concepts, and the environments that promote or inhibit learning. While computer science education has made progress toward outlining a learning trajectory for introductory programming concepts, analogous research focusing on how students build skills related to computing with data has yet to be pursued. Additionally, these investigations of student learning could probe how differences in the `R` programming environment contribute to students' understanding of data science concepts, specifically between base `R` and the `tidyverse`. For example, novice programmers may find it difficult to understand a data wrangling process associated with filtering a dataset, selecting columns, and mutating variables, when the entire process is coded in a single line of `R` code. Instead, using `dplyr` (Wickham et al., 2018) to break each of these data wrangling steps apart may have a substantial impact on students' learning. Finally, as the number of universities requiring general computing courses for undergraduates in scientific disciplines continues to increase (Cortina, 2007; Rubinstein and Chor, 2014; Wilson et al., 2008), a focused investigation into how students transfer general programming knowledge to their understanding of data

science concepts can leverage intracurricular understandings.

Conclusion

Alicia's experiences acquiring the data science skills necessary to implement statistics in her research echo the sentiment heard throughout the environmental science literature. A single statistics course, peers, and the internet were the pathways Alicia attributed to learning the computing skills she employed in her research. While Alicia's experiences reiterate the computational ill-preparation of environmental science graduate students by their curriculum, Ellie's experiences add a perspective yet to be acknowledged by the environmental science education literature. Where Alicia struggled to claim a basic understanding of `R`, Ellie's data science skills flourished. Ellie's flexibility in data structures and fluency in the `R` environment accelerated her ability to acquire new data science skills, and afforded her research opportunities out of the reach of most graduate students. Ellie's abilities and experiences add a voice to the environmental science graduate students in Hernandez et al.'s survey (2012) that reported an expert proficiency in any given programming language.

Because many studies have focused on outlining students' lack of preparation for data-intensive research, no attention has been paid to understanding the experiences of students who do leave their curriculum with these critical skills. Ellie's extensive experiences programming in MATLAB for her undergraduate engineering degree, her use of an interactive tool for learning `R` (Kross et al., 2018), and the support of her adviser and committee member provide a window into the experiences which supported Ellie growing her data science skills. Ellie's undergraduate experiences reiterate the importance of students from every scientific discipline leaving their program with data science skills relevant to their daily lives. Furthermore, the support Ellie received from her adviser, suggesting resources to grow her `R` proficiency, emphasizes the potential impact of advisers on the learning trajectory of their students.

Disappointingly, outside of outlining the "good enough" practices in scientific computing (Wilson et al., 2017), no research has sought to understand what specific data science skills are necessary for graduate students in the environmental sciences as they engage in data-intensive research. In this paper, we contribute to the research in this field by describing the key data science skills used by two environmental science graduate students throughout their data analysis cycle. While these skills are not exhaustive, they add a perspective not yet heard in the statistics education or environmental science literature: the data science skills necessary for graduate-level *practitioners* of statistics in environmental science fields.

The computational topics relevant to statistics outlined by Nolan & Temple Lang (2010) share little overlap with the data science skills used by both Alicia and Ellie. The overlap in these skills is encompassed primarily by the statistical computing topics of reproducible computation and data structures, and the visualization topics of the grammar of graphics and color. Meanwhile, the taxonomy of skills for data-intensive environmental science research (Hampton et al., 2017) potentially shares a greater overlap with the data science skills used by both Alicia and Ellie, but the vague nature of the taxonomy makes direct comparisons difficult. Throughout their entire data analysis cycle, both Ellie and Alicia used `R` scripts for data analysis, and it could be argued they also used "the fundamentals of data management," "data transform," and "visual literacy and graphical principles" (p. 549). This suggests that a substantial number of the data science skills outlined by both fields may not be necessary understandings for environmental science graduate students as they participate in data-intensive research.

Nearly every computing practice Wilson and colleagues (2017) outlined as essential for every scientific researcher appeared in the `R` scripts generated by Alicia and Ellie. The importance of many of these practices was seen throughout Alicia's research code, including, but not limited to, reproducible data processing, eliminating duplication, code commenting to control a script's behavior, project working directories, and naming objects

to reflect their contents. Alicia's lack of formal statistics and computing education and support continually hindered her ability to acquire new data science skills. Wilson and colleagues' statement that these skills are comfortably within reach of "every researcher," brings us to question: Why are students leaving their graduate program without these fundamental skills? Furthermore, how can we, as educators of scientific practitioners of statistics, ensure that students leave our courses with the data science skills needed to wholeheartedly participate in data-intensive environmental science research?

Acknowledgements

We would like to specially thank the participants from this study, without whom this research would not have been possible. We thank you for your courage, submitting personal artifacts of your research journey and speaking openly about your experiences. We would also like to thank Mary Alice Carlson, Jennifer Green, Mark Greenwood, and Megan Wickstrom for their insightful comments on this paper.

CONCLUSION

This body of research is intended to provoke thought and discussion surrounding the computational preparation of graduate students in the environmental sciences with the data science skills necessary to engage in the entire cycle of data analysis. We began by outlining the evolution of the fields of statistics and the environmental sciences, catalyzed by the rapid increase in the volume and variety of data, and the computational tools available for analysis. These dramatic changes to the data landscape created a crucial need to re-evaluate the preparation of scientific researchers. Calls for this revitalization were echoed across both statistics and the environmental sciences, yet environmental science researchers continue to report that students are are not learning these critical skills in their curriculum (Hampton et al., 2017; Hernandez et al., 2012; Teal et al., 2015).

While Hernandez et al. outlined the shape of this ill preparation by describing the coursework and topics that students reported never encountering in their graduate program, no study had focused on the experiences of graduate students acquiring the computing skills necessary to analyze their data. This gap in knowledge inspired me to investigate how environmental science graduate students experience the phenomenon of acquiring the computing skills necessary to implement statistics in the context of their research. Through in-depth interviews with five environmental science graduate students, we uncovered three themes in students' paths to computational knowledge acquisition: use of peer support, seeking out a singular "consultant," and learning through independent research experiences. Furthermore, we described the statistical computing skills students reported leaving the statistics classroom with, and how their backgrounds affected their experiences acquiring these necessary skills. By in large, these students reported learning the computing skills necessary to analyze their data on their own or through information that was passed down within their social network. As suggested by statistics and environmental science educators alike, this "do it yourself" system results in substantial hidden costs and impedes the progress of scientific research (Nolan and Temple Lang, 2010; Teal et al., 2015). Because

the computing included in the environmental science curriculum continues to lag behind, environmental science educators have repeatedly recommended extracurricular workshops as a bridge for students to acquire the foundational data science skills needed to conduct research (Hampton et al., 2017; Hernandez et al., 2012; Teal et al., 2015; Wilson, 2006).

The computational ill preparation of environmental science graduate students by their curriculum leaves a need for high-quality, relevant, and accessible trainings, equipping students with the data science skills needed to conduct research in their field. Although Data Carpentry workshops offer domain-specific training intended to provide researchers with the foundational skills necessary for data-driven research (Data Carpentry, 2020), no attention had been paid to the relevance of the content of these workshops to specific populations of researchers. This need motivated me to investigate how these discipline-specific workshops could be tailored to meet the needs of environmental science graduate students. However, this investigation required a more comprehensive understanding of the computing skills necessary for environmental science graduate students throughout their data analysis cycle.

Through interviews with environmental science faculty, we learned that faculty believed students need extensive experiences working with data and visualizing data, both using reproducible tools. Additionally, these faculty reiterated sentiments heard throughout the environmental science literature, that these students are not learning the data skills necessary for their research in the coursework required for their degree. The foundational data skills outlined by these faculty were then infused into Data Carpentry's *Data Analysis and Visualization in `R` for Ecologists* lesson (Michonneau et al., 2019). In addition, faculty also outlined data skills, such as conditional statements and repeated operations, that were not currently included in this Data Carpentry lesson. Software Carpentry, however, offers a lesson for learning to program in `R`, which teaches many of these additional programming skills (Wright and Zimmerman, 2016). Thus, these additional skills were integrated into the *`R` for Reproducible Scientific Analysis* lesson, tailored to have the same environmental

science context as the other workshops. In the end, a suite of four workshops were developed: *Introduction to* `R`, *Intermediate* `R`, *Data Wrangling with* `dplyr` *and* `tidyr`, and *Data Visualization with* `ggplot2`.

During the 2018-2019 academic year, this suite of workshops was offered through a partnership with the Montana State University library. Advertised across campus, a total of 202 students, faculty, and staff attended at least one of the workshops. Although, many of the attendees solely attended the *Introduction to* `R` workshop, many attendees still selected to return for subsequent workshops. The attendees' pre-workshop surveys were consistent with what had been heard in the literature, as over 75% of workshop attendees had completed no formal courses in computer programming (Andelman et al., 2004; Hampton et al., 2017; Hernandez et al., 2012; Teal et al., 2015). Additionally, we discovered that the preponderance of these attendees had only taken a single statistics course, covering introductory concepts. As might be expected from the prevalent use of `R` in environmental science research (Lai et al., 2019; Mislan et al., 2016) and the current state of computing in the environmental science curriculum, over half of the master's and doctoral workshop participants attended the workshops for assistance with their research. Furthermore, the majority of these attendees reported using the internet and their peer networks as the main resource for learning `R`, consonant with the suspicions of environmental science educators (Teal et al., 2015). Finally, numerous participants reported that, at the workshop, they were hoping to learn something related to analyzing data, a desire which reiterates the importance of data science skills throughout the data analysis cycle.

The iterative nature of design-based implementation research demands the researcher revisit the content of their teaching innovation, to reassess its alignment with the desired learning outcomes. As the goal of these workshops is to equip environmental science graduate students with the data science skills necessary to conduct their research, this reevaluation of the workshop content requires an understanding of the data science skills these students are actually using in their research. Paired with the need to distill the

broad classes of computing skills outlined by statistics and environmental science educators (Hampton et al., 2017; Nolan and Temple Lang, 2010), I embarked on research which would illuminate these foundational skills.

Case study research allows for the investigation of "a contemporary phenomenon within its real-life context," (Yin, 2009, p. 18) when the boundaries between the phenomenon and the context are not easily deciphered. Where the prevalence of the phenomenon of environmental science graduate students acquiring the computing skills necessary for their research has been outlined by environmental science educators (Hampton et al., 2017; Hernandez et al., 2012; Strasser and Hampton, 2012; Teal et al., 2015) and the first arm of this research (Theobold and Hancock, 2019), none of these studies have sought to understand this phenomenon in the context of students and their research. To illuminate the data science skills environmental science students use throughout the data analysis cycle, we conducted an embedded, comparative case study. By analyzing the research code generated by two environmental science graduate students, Alicia and Ellie, we identified themes of data science skills each student used throughout their code and created concept maps outlining the interwoven nature of these skills. This longitudinal exploration of the data science skills used by each of these women allowed us to map how each student's skills evolved over time. Furthermore, interviews with these women regarding their experiences acquiring the data science skills they made use of adds new perspectives to the discussions surrounding the computational preparation of graduate students in the environmental sciences by contrasting the computational preparation and support Ellie experienced with that of Alicia.

Rather than generating extensive evidence of the need for training for environmental science graduate students in the computing skills related to data, our research has focused on describing and understanding the nature of this need, through the voices of the graduate students. Our research has described these students' experiences acquiring the computing skills necessary to implement statistics in their research, explored how extracurricular

workshops can be tailored to meet the needs of this population of researchers, and began the work toward identifying key data science skills necessary for these students as they engage in the data analysis cycle. Moreover, this research attests to the inseparable nature of statistics and data science, consistently focusing on the data science skills necessary for environmental science graduate students as they endeavor to implement the statistical analyses dictated by their research.

Directions for Future Research

Data science is here to stay—giving a name to the computing skills necessary for researchers to engage in the entire data analysis cycle. Potentially reflecting the growing awareness that statistical analyses are but one piece in the puzzle, the Google searches for "data science" now greatly overshadow that of "statistics" (Figure 5.1). However, this dramatic shift has left statistics and environmental science educators grappling with what data science topics belong in the curriculum, when to teach them, and how they should be taught.

While these disciplines have extensively outlined data science topics of potential relevance to researchers in their respective field, there currently is no understanding of how students learn concepts in data science, how these concepts build on each other, and what understandings foster or inhibit new learning. Research outlining a learning trajectory for data science concepts should be of utmost concern to the discipline of statistics education. The beginnings of this research can be seen as Alicia acquired the ability to filter rows of her data using a variety of tools, but lacked the ability to synthesize how each tool could solve a broader array of data tasks. Would Alicia's understanding of selecting columns have fostered her understanding of how to filter her data? Or could this understanding have been built from a fluency with data structures?

Statistics educators are in a position of great responsibility to communicate how to appropriately teach data science concepts. With data science growing in popularity, we

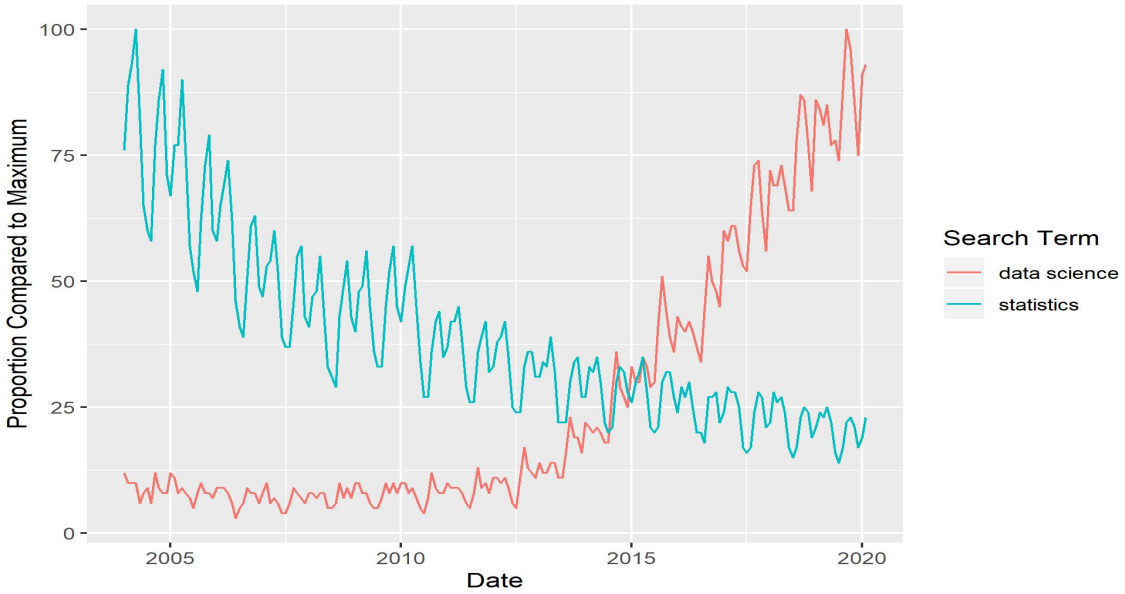


Figure 5.1: Google trends for search terms "data science" and "statistics" as of February 23, 2020. The y-axis represents search interest relative to the highest point on the chart between 2004 and 2020, where 100 is the peak popularity for the term.

need to remind researchers that statistics is more than data analysis. Rather, statistics, like data science, encompasses the entire process of "extracting value from data" (Wing, 2019). We hope to have provided environmental science researchers with an understanding and appreciation for the computing skills necessary for graduate students to implement statistics, while also emphasizing to statistics educators the importance of incorporating data science concepts into *every* statistics course.

REFERENCES CITED

Bibliography

Altadmri, A. and Brown, N. C. (2015). 37 million compilations: Investigating novice programming mistakes in large-scale student data. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 522–527. ACM.

American Statistical Association Undergraduate Guidelines Workgroup (2014). *2014 curriculum guidelines for undergraduate programs in statistical science.* American Statistical Association, Alexandria, VA.

Andelman, S. J., Bowles, C. M., Willig, M. R., and Waide, R. B. (2004). Understanding environmental complexity through a distributed knowledge network. *BioScience*, 54(3):240–246.

Baumer, B. (2015). A data science course for undergraduates: Thinking with data. *The American Statistician*, 69(4):334–342.

Baumer, B., Cetinkaya-Rundel, M., Bray, A., Loi, L., and Horton, N. J. (2014). Use of R as a toolbox for mathematical statistics exploration. *Technology Innovations in Statistics Education*, 8(1):1–30.

Baumer, B. S., Horton, N. J., and Wickham, H. (2015). Setting the stage for data science: Integration of data management skills in introductory and second courses in statistics. *CHANCE*, 28(2):40–50.

Bernard, H. R. (1988). *Research Methods in Cultural Anthropology.* Sage Publications, Inc., Newbury Park, California.

Biehler, R. (1997). Software for learning and for doing statistics. *International Statistical Review*, 62(2):167–189.

Bilkstein, P. (2011). Using learning analytics to assess students' behavior in open-ended programming tasks. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, pages 110–116. ACM.

Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231.

Brown, E. N. and Kass, R. E. (2009). What is statistics? *The American Statistician*, 63:105–110.

Brown, N. C. C. and Altadmri, A. (2014). Investigating novice programming mistakes: Educator beliefs vs. student data. In *Proceedings of the 10th Annual Conference on International Computing Education Research*, pages 43–50. ACM.

Bryce, G. R., Gould, R., Notz, W. I., and Peck, R. L. (2001). Curriculum guidelines for bachelor of science degrees in statistical science. *The American Statistician*, 55(1):7–13.

Bulmer, J., Pinchbeck, A., and Hui, B. (2018). Visualizing code patterns in novice programmers. In *23rd Western Canadian Conference on Computing Education.* ACM.

Caceffo, R., Wolfman, S., Booth, K. S., and Azevedo, R. (2016). Developing a computer science concept inventory for introductory programming. In *Proceedings of the 47th ACM Technical Symposium on Computer Science Education*, pages 364–369. ACM.

Cannon, A., Hartlaub, B., Lock, R., Notz, W., and Parker, M. (2002). Guidelines for undergraduate minors and concentrations in statistical science. *Journal of Statistics Education*, 10(2).

Cassey, P. and Blackburn, T. M. (2006). Reproducibility and repeatability in ecology. *BioScience*, 56(12):98.

Cetinkaya-Rundel, M. (2018). Intro stats, intro data science: Do we need both? Presented at the 2018 Joint Statistical Meetings.

Cetinkaya-Rundel, M. and Rundel, C. (2018). Infrastructure and tools for teaching computing throughout the statistical curriculum. *The American Statistician*, 72(1):58–65.

Chang, W. (2019). *R6: Encapsulated Classes with Reference Semantics*. R package version 2.4.0.

Cherenkova, Y., Zingaro, D., and Petersen, A. (2014). Identifying challenging CS1 concepts in a large problem dataset. In *Proceedings of the 47th ACM Technical Symposium on Computer Science Education*, pages 695–700. ACM.

Cleveland, W. S. (2001). Data science: An action plan for expanding the technical areas of the field of statistics. *International Statistical Review*, 69(1):21–26.

Cobb, G. (2015). Mere renovation is too little too late: We need to rethink our undergraduate curriculum from the ground up. *The American Statistician*, 69(4):266–282.

Cobb, P. A., Confrey, J., diSessa, A. A., Lehrer, R., and Schauble, L. (2003). Design experiments in educational research. *Educational Researcher*, 32(1):9–13.

Cortina, T. J. (2007). An introduction to computer science for non-majors using principles of computation. *SIGCSE Bull.*, 39(1):218–222.

Creswell, J. (2007). *Qualitative inquiry and research design: Choosing among five approaches*. Sage Publications, Inc., 2nd edition.

Data Carpentry (2020). https://datacarpentry.org/.

Denzin, N. (1978). *Sociological Methods*. McGraw-Hill, New York.

Dodds, Z., Alvarado, C., Kuenning, G., and Libeskind-Hadas, R. (2007). Breadth-first CS 1 for scientists. *ACM SIGCSE Bulletin*, 39(3):23–27.

Dodds, Z., Libeskind-Hadas, R., Alvarado, C., and Kuenning, G. (2008). Evaluating a breadth-first CS 1 for scientists. *ACM SIGCSE Bulletin*, (1):266–270.

Eglen, S. J. (2009). A quick guide to teaching R programming to computational biology students. *PLOS Computational Biology*, 5(8):1–4.

Ellison, A. M. (2010). Repeatability and transparency in ecological research. *Ecology*, 91(9):2536–2539.

Ernest, M., Brown, J., Valone, T., and White, E. P. (2018). Portal project teaching database.

Fishman, B. J., Penuel, W. R., Allen, A.-R., Cheng, B. H., and Sabelli, N. (2013). Design-based implementation research: An emerging model for transforming the relationship of research and practice. In Fishman, B. J. and Penuel, W. R., editors, *Design Based Implementation Research*, volume 112, pages 136–156. National Society for the Study of Education.

Fox, J. A. and Ouellette, B. F. (2013). Education in computational biology today and tomorrow. *PLOS Computational Biology*, 9(12):1–2.

Friedman, J. (2001). The role of statistics in the data revolution. *International Statistics Review*, 69:5–10.

Gould, R. (2010). Statistics and the modern student. *International Statistics Review*, 78:297–315.

Green, J. L. and Blankenship, E. E. (2015). Fostering conceptual understanding in mathematical statistics. *The American Statistician*, 69(4):315–325.

Green, J. L., Hastings, A., Arzberger, P., Ayala, F. J., Cottingham, K. L., Cuddington, K., Davis, F., Dunne, J. A., Fortin, M.-J., Gerber, L., and Neubert, M. (2005). Complexity in ecology and conservation: Mathematical, statistical, and computational challenges. *BioScience*, 55(6):501–510.

Grimshaw, S. D. (2015). A framework for infusing authentic data experiences within statistics courses. *The American Statistician*, 69(4):307–314.

Gutlerner, J. L. and Van Vactor, D. (2013). Catalyzing curriculum evolution in graduate science education. *Cell*, 153(4):731–736.

Hambrusch, S., Hoffman, C., Korb, J. T., Haugan, M., and Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. In *Proceedings of the 2009 SIGCSE*, pages 183–187. ACM.

Hampton, S. E., Jones, M. B., Wasser, L. A., Schildhauer, M. P., Supp, S. R., Brun, J., Hernandez, R. R., Boettiger, C., Collins, S. L., Gross, L. J., Fernandez, D. S., Budden, A., White, E. P., Teal, T. K., Labou, S. G., and Aukema, J. E. (2017). Skills and knowledge for data-intensive environmental research. *BioScience*, 67(6):546–557.

Hardin, J. (2018). Dynamic data in the statistics classroom. *Technology Innovations in Statistics Education*, 11(1):1–22.

Hardin, J., Hoerl, R., Horton, N. J., Nolan, D., Baumer, B., Hall-Holt, O., Murrell, P., Peng, R., Roback, P., Temple Lang, D., and Ward, M. D. (2015). Data science in statistics curriculua: Preparing stuents to "think with data". *The American Statistician*, 69(4):343–353.

Hastings, A., Arzberger, P., Bolker, B., Collins, S., Ives, Anthony, R., Johnson, N. A., and Palmer, M. A. (2005). Quantitative bioscience for the 21st century. *BioScience*, 55(6):511–517.

He, X., Madigan, D., Yu, B., and Wellner, J. (2019). Statistics at a crossroads: Who is for the challenge. Technical report, The National Science Foundation.

Hernandez, R. R., Mayernik, M. S., Murphy-Mariscal, M. L., and Allen, M. F. (2012). Advanced technologies and data management practices in environmental science: Lessons from academia. *BioScience*, 62(12):1067–1076.

Hesterberg, T. C. (2015). What teachers should know about the bootstrap: Resampling in the undergraduate statistics curriculum. *The American Statistician*, 69(4):371–386.

Higgins, J. J. (1999). Nonmathematical statistics: A new direction for the undergraduate discipline. *The American Statistician*, 53(1):1–6.

Hodder, I. (1994). The interpretation of documents and material culture. In Denzin, N. K. and Lincoln, Y. S., editors, *Handbook of qualitative research*, pages 393–402. Sage Publications, Inc., Thousand Oaks, California.

Horton, N. J., Brown, E. R., and Qian, L. (2004). Use of R as a toolbox for mathematical statistics exploration. *The American Statistician*, 58(4):343–357.

Horton, N. J. and Hardin, J. S. (2015). Teaching the next generation of statistics students to "think with data": Special issue on statistics and the undergraduate curriculum. *The American Statistician*, 69(4):259–265.

Hristova, M., Misra, A., Rutter, M., and Mercuri, R. (2003). Identifying and correcting Java programming errors for introductory computer science students. In *Proceedings of the 34th ACM Technical Symposium on Computer Science Education*, pages 153–156. ACM.

Johnson, G. (2001). The world: In silica fertilization; all science is computer science. *New York Times*.

Johnson, G. (2014). New truths that only one can see. *The New York Times*.

Jones, M. B., Schildhauer, M. P., Reichman, O., and Bowers, S. (2006). The new bioinformatics: Integrating ecological data from the gene to the biosphere. *Annual Review of Ecology, Evolution, and Systematics*, 37(1):519–544.

Joppa, L. N., McInerny, G., Harper, R., Salido, L., Takeda, K., O'Hara, K., Gavaghan, D., and Emmott, S. (2013). Troubling trends in scientific software use. *Science*, 340(6134):814–815.

Kaplan, D. (2018). Teaching stats for data science. *The American Statistician*, 72(1):89–96.

Kelling, S., Hochachka, W. M., Fink, D., Riedewald, M., Caruana, R., Ballard, G., and Hooker, G. (2009). Data-intensive science: A new paradigm for biodiversity studies. *BioScience*, (59):613–620.

Keon-Woong, M. (2019). *ggiraphExtra: Make Interactive 'ggplot2'. Extension to 'ggplot2' and 'ggiraph'.* R package version 0.2.9.1.

Kitzes, J., Turek, D., and Deniz, F. (2018). *The Practice of Reproducible Research: Case Studies and Lessons from the Data-Intensive Sciences.* University of California Press, Okland, CA. Available from: https://www.practicereproducibleresearch.org/.

Kross, S., Carchedi, N., Bauer, B., Grdina, G., Schouwenaars, F., and Wu, W. (2018). *swirl: Learn R, in R.* R package version 2.4.3.

Lahtinen, E., Ala-Mutka, K., and Jarvinen, H. M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, 37(3):14–18.

Lai, J., Lortie, C. J., Muenchen, R. A., Yang, J., and Ma, K. (2019). Evaluating the popularity of R in ecology. *Ecosphere*, 10(1).

Laney, C. M., Pennington, D. D., and Tweedie, C. E. (2015). Filling the gaps: sensor network use and data-sharing practices in ecological research. *Frontiers in Ecology and the Environment*, 13(7):363–368.

Levin, S. A., Grenfell, B., Hastings, A., and Perelson, A. S. (1977). Mathematical and computational challenges in popluation biology and ecosystems science. *Science*, 275(5298):334–343.

Lock, R., Lock, P. F., Lock Morgan, K., Lock, E. F., and Lock, D. F. (2013). *Unlocking the Power of Data.* Wiley, Hoboken, New Jersey.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., and Byers, A. H. (2011). Big data: The next frontier for innovation, competition, and productivity. Technical report, McKinsey Global Institute.

McNamara, A. and Horton, N. (2018). Wrangling categorical data in R. *The American Statistician*, 72(1):97–104.

Merriam, S. B. (2009). *Qualitative research, a guide to design and implementation.* Jossey-Bass, 3rd edition.

Michonneau, F., Teal, T., Fournier, A. M., Seok, B., and Conrado, A. C. (2019). Data carpentry: Data analysis and visualization in R for ecologists.

Miles, M. B. and Huberman, A. M. (1994). *Qualitative Data Analysis, An Expanded Sourcebook.* Sage Publications, Inc., Thousand Oaks, California.

Miles, M. B., Huberman, A. M., and Saldana, J. (2014). *Qualitative Data Analysis, A Methods Sourcebook*. Sage Publications, Inc., Thousand Oaks, California, 3rd edition.

Milne, I. and Rowe, G. (2002). Difficulties in learning and teaching programming views of students and tutors. *Education and Information Technologies*, 7(1):55–66.

Mislan, K., Heer, J., and White, E. (2016). Elevating the status of code in ecology. *Trends in Ecology & Evolution*, 31(1):4–7.

Mokany, K., Ferrier, Simon amd Connolly, S. R., Dunstan, P. K., Fulton, E. A., Harfoot, M. B., Harwood, T. D., Richardson, A. J., Roxburgh, S. H., Scharlemann, J. P. W., Tittensor, D. P., Westcott, D. A., and Wintle, B. A. (2016). Integrating modelling of biodiversity composition and ecosystem function. *Oikos*, 125(1):10–19.

Moore, D. S., Cobb, G. W., Garfield, J., and Meeker, W. Q. (1995). Statistics education fin de siecle. *The American Statistician*, 49(3):250–260.

Moreno, J. L. (2002). Toward a statistically literate citizen: What statistics everyone should know. In *Proceedings of the 6th International Conference on Teaching Statistics*. IASE.

Morrison, C., Wardle, C., and Castley, J. (2016). Repeatability and reproducibility of population viability analysis (pva) and the implications for threatened species management. *Frontiers in Ecology and Evolution*, 4:98.

Moustakas, C. (1994). *Phenomenological research methods*. Sage Publications, Inc.

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

National Research Council (1994). *Modern Interdisciplinary University Statistics Education: Proceedings of a Symposium*. The National Academies Press, Washington, DC.

Newman, H. B., Ellisman, M. H., and A., O. J. (2003). Data-intensive e-science frontier research. *Communications of the ACM*, 46(11):68–77.

Nolan, D. and Perrett, J. (2016). Teaching and learning data visualization: Ideas and assignments. *The American Statistician*, 70(3):260–269.

Nolan, D. and Speed, T. (2000). *Stat Labs: Mathematical Statistics Through Applications*. Springer, New York.

Nolan, D. and Speed, T. P. (1999). Teaching statistics theory through applications. *The American Statistician*, 53(4):370–375.

Nolan, D. and Temple Lang, D. (2010). Computing in the statistics curricula. *The American Statistician*, 64(2):97–107.

Nolan, D. and Temple Lang, D. (2015). Explorations in statistics research: An approach to expose undergraduates to authentic data analysis. *The American Statistician*, 69(4):292–299.

O'Neill, D. K. (2012). Designs that fly: What the history of aeronautics tells us about the future of design-based research in education. *International Journal of Research and Method in Education*, 35(2):119–140.

Patton, M. Q. (2002). *Qualitative research and evaluation methods*. Sage Publications, Inc., 3rd edition.

Peck, R. and Chance, B. (2005). Assessing effectiveness and the program level: Undergraduate statistics program evaluation. In *Proceedings of the 2005 Joint Statistics Meetings*. American Statistical Association.

Peters, D. and Okin, G. (2017). A toolkit for ecosystem ecologists in the time of big science. *Ecosystems*, 20:259–266.

Petre, M. and van der Hoek, A. (2016). *Software Design Decoded: 66 Ways Experts Think*. MIT Press.

Powers, S. M. and Hampton, S. E. (2019). Open science, reproducibility, and transparency in ecology. *Ecological Applications*, 29(1).

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. R version 4.0.0.

Ramsey, F., modifications by Daniel W. Schafer, D. S., Sifneos, J., vignettes contributed by Nicholas Horton, B. A. T., Loi, L., Aloisio, K., Zhang, R., and with corrections by Randall Pruim (2019). *Sleuth3: Data Sets from Ramsey and Schafer's "Statistical Sleuth (3rd Ed)"*. R package version 1.0-3.

Reid, N., Efron, B., and Morris, C. (2003). Is the math stat course obsolete?

Revision Committee, A. (2014). *Guidelines for Assessment and Instruction in Statistics Education College Report 2016*. American Statistical Association, Alexandria, VA.

Ross, Z., Wickham, H., and Robinson, D. (2017). Declutter your R workflow with tidy tools. Technical report, PeerJ Preprints.

Rossman, A. J. and Chance, B. L. (2011). *Workshop Statistics*. Wiley, Hoboken, New Jersey.

RStudio Team (2015a). *RStudio Cloud*. RStudio, Inc., Boston, MA.

RStudio Team (2015b). *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA.

Rubinstein, A. and Chor, B. (2014). Computational thinking in life science education. *PLOS Computational Biology*, 10(11):1–5.

Sandve, G. K., Nekrutenko, A., Taylor, J., and Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLOS Computational Biology*, 9(10):1–4.

Sarkar, D. (2008). *Lattice: Multivariate Data Visualization with R*. Springer, New York.

Schram, T. A. (2003). *Conceptualizing qualitative inquiry*. Merrill Prentice Hall, 3rd edition.

Smith, D. (2015). Vision and change in undergraduate biology education: Chronicling change, inspiring the future. Technical report, American Association for the Advancement of Science.

Software Carpentry (2020). https://software-carpentry.org/.

Stake, R. E. (2006). *Multiple Case Study Analysis*. The Guilford Press, New York.

Stefan, M. I., Gutlerner, J. L., Born, R. T., and Springer, M. (2015). The quantitative methods boot camp: Teaching quantitative thinking and computing skills to graduate students in the life sciences. *PLOS Computational Biology*, 11(4):1–12.

Strasser, C. A. and Hampton, S. E. (2012). The fractured lab notebook: Undergraduates and ecological data management training in the united states. *Ecosphere*, 3(12):1–18.

Teal, T. K., Cranston, K. A., Lapp, H., White, E., Wilson, G., Ram, K., and Pawlik, A. (2015). Data carpentry: Workshops to increase data literacy for researchers. *International Journal of Digital Curation*, 10(1):135–143.

The Carpentries (2019). https://carpentries.org/.

The Economist Editorial (2013). *Trouble at the lab. (Cover story)*. .

Theobold, A. and Hancock, S. (2019). How environmental science graduate students acquire statistical computing skills. *Statistics Education Research Journal*, 18(2):68–85.

Tintle, N., Chance, B., Cobb, G., Roy, S., Swanson, T., and VanderStoep, J. (2015). Combating anti-statistical thinking using simulation-based methods throughout the undergraduate curriculum. *The American Statistician*, 69(4):362–370.

Tukey, J. (1962). The future of data analysis. *Annals of Statistics*, 33(1):1–67.

Utts, J. (2003). What educated citizens should know about statistics and probability. *The American Statistician*, 57(2):74–79.

Van Manen, M. (1990). *Researching lived experience: Human science for an action sensitive pedagogy*. State University of New York.

Wang, X., Rush, C., and Horton, N. J. (2017). Data visualization on day one: Bringing big ideas into intro stats early and often. *Technology Innovations in Statistics Education*, 10(1):1–22.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., and Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1):127–147.

Wickham, H. (2011). The split-apply-combine strategy for data analysis. *Journal of Statistical Software*, 40(1):1–29.

Wickham, H. (2014). Tidy data. *The Journal of Statistical Software*, 59(10).

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York.

Wickham, H. (2017). *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.2.1.

Wickham, H. (2019). *Advanced R*. Chapman & Hall, Boca Raton, Florida, 2nd edition.

Wickham, H., Francois, R., Henry, L., and Muller, K. (2018). *dplyr: A Grammar of Data Manipulation*. R package version 0.7.6.

Wickham, H. and Grolemund, G. (2017). *R for Data Science*. O'Reilly, Sebastopol, California.

Wickham, H., Hester, J., and Chang, W. (2019). *devtools: Tools to Make Developing R Packages Easier*. R package version 2.2.1.

Wilkinson, L. (2005). *The Grammar of Graphics*. Springer, Hoboken, New Jersey, 2nd edition.

Wilson, G. (2006). Software carpentry: Getting scientists to write better code by making them more productive. *Computing in Science & Engineering*, 8(6):66–69.

Wilson, G. (2016). Software carpentry: lessons learned. *F1000 Research*, 3(62).

Wilson, G. (2019). *Teaching Tech Together: How to Make your lessons work and build a teaching community around them*. Chapman and Hall, Boca Raton, Florida.

Wilson, G., Alvarado, C., Campbell, J., Landau, R., and Sedgewich, R. (2008). CS-1 for scientists. In *Technical Symposium with Computer Science Education*, pages 36–37. ACM.

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(6):1–20.

Wing, J. (2006). Computational thinking. *Communications of ACM*, 49(3):33–35.

Wing, J. M. (2019). The data life cycle. *Harvard Data Science Review*, 1(1).

Word, K. R., Jordan, K., Becker, E., Williams, J., Reynolds, P., Hodge, A., Belkin, M., Marwick, B., and Teal, T. (2017). When do workshops work? A response to the 'null effects' paper from Feldon et al. Technical report, Software Carpentry.

Worsley, M. and Blikstein, P. (2013). Programming pathways: A technique for analyzing novice programmers learning trajectories. In Lane, H., Yacef, K., Mostow, J., and Pavlik, P., editors, *Artificial Intelligence in Education*. AIED 2013.

Wright, T. and Zimmerman, N. (2016). Software carpentry: R for reproducible scientific analysis.

Yin, R. K. (2009). *Case Study Research: design and methods*. Sage Publications, Inc.

APPENDICES

APPENDIX A

STATISTICAL COMPUTING TASKS FROM CHAPTER TWO

We have data on fish caught in the Blackfoot River by Fish, Wildlife, & Parks personnel over a number of years. They used electrofishing equipment to attract the fish to the boat, then dipped them out of the water with nets, measured length in cm and weight in grams. They are often working in cold conditions in late autumn or early spring, so some measurement error is expected.

These data are not from a random sample. The goal is to catch all fish within a reach or section of the Blackfoot River every few years to assess the health of the population. Changes over years are important to the biologists.

The data were collected by making two trips per section (Johnsrud or Scotty Brown) each sampling year. The fish caught each trip of a given year, had their weight, length, and species recorded.

```
head(blackfoot)

##   trip length weight year  section species
## 1    1    288    175 1989 Johnsrud     RBT
## 2    1    288    190 1989 Johnsrud     RBT
## 3    1    285    245 1989 Johnsrud     RBT
## 4    1    322    275 1989 Johnsrud     RBT
## 5    1    312    300 1989 Johnsrud     RBT
## 6    1    363    380 1989 Johnsrud     RBT

summary(blackfoot)

##       trip         length        weight          year
##  Min.   :1.0   Min.   : 16   Min.   :   0   Min.   :1989
##  1st Qu.:1.0   1st Qu.:186   1st Qu.:  65   1st Qu.:1991
##  Median :2.0   Median :250   Median : 150   Median :1996
##  Mean   :1.5   Mean   :262   Mean   : 246   Mean   :1997
##  3rd Qu.:2.0   3rd Qu.:330   3rd Qu.: 330   3rd Qu.:2002
##  Max.   :2.0   Max.   :986   Max.   :4677   Max.   :2006
##                              NA's   :1796
##    section            species
##  Length:18352       Length:18352
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

```
str(blackfoot)

## Observations: 18,352
## Variables: 6
## $ trip    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ length  <dbl> 288, 288, 285, 322, 312, 363, 269, 160, 213, ...
## $ weight  <dbl> 175, 190, 245, 275, 300, 380, 170, 40, 80, ...
## $ year    <dbl> 1989, 1989, 1989, 1989, 1989, 1989, 1989, ...
## $ section <chr> "Johnsrud", "Johnsrud", "Johnsrud", ...
## $ species <chr> "RBT", "RBT", "RBT", "RBT", "RBT", "RBT", ...
```

- What type of variable did `R` store `species` and `section` as? How would you change species and section to categorical variables?
- If the researchers were only interested in Rainbow trout and Brown trout, how would you remove Bull trout and WCT (whitefish) from the data set?
- Sometimes when sampling the fish, a technician fails to record one of the variables. How would you remove all the fish with missing values? How would this change if you instead removed the fish with only missing weight?
- The sampling methods used by Fish, Wildlife, & Parks on the Blackfoot River has changed over the years. In the years 1989 - 1996 they used gill nets and since 1996 they have used electrofishing. How would you create a new variable named `method` to reflect these different sampling methods used over the years?
- The researchers are interested in how many fish are caught each year that weigh over 1500 grams. How would you find these numbers to report?
- Which pairs of (weight, length) combinations seem difficult to believe? One way to look for unusual pairs is to use what fisheries biologists call a "condition index": $\frac{w^{1/3}}{l} \times 50$, where w = weight and l = length of the fish. If fish are highly unusual in this scale, it would be best to remove them, but you might need to compare only within species.
- How would you calculate each trout's condition number?
- How would you summarize these condition numbers for each of the two species of trout (Rainbow and Brown)?
- How would you plot the condition numbers of each trout, making sure to differentiate between Rainbow and Brown trout?
- The researchers are interested in trends in fish size over the sampling period (1989-2006). How would you create a visualization of fish lengths over the sampling period?

- Researchers are also interested in the number of fish from each species caught each year. How would you create a visualization of the number of fish caught from each species over the sampling period?

Lastly, the researchers are interested in trends in average fish weight over the sampling period. They want you to create a visualization of the average fish weight across years, differentiated by species of trout.

- First, you need to create a data frame of the mean weight of fish caught each year for the two species of trout. The end product should look something like the data frame below. How would you create this data frame of mean weights?

```
##   year species mean
## 1 1989   Brown  297
## 2 1989    Bull  429
## 3 1989     RBT  101
## 4 1989     WCT  120
## 5 1990   Brown  380
## 6 1990    Bull  422
```

- Next, to plot these mean weights for each year you need to transform the data from the current long format to wide format. This process is done by spreading the year variable across 10 different columns, one for each year (1989, 1990, etc.). The end product should look something like the data frame below. How would you transform these data from long format to wide format?

```
##   species 1989 1990 1991 1993 1996 1998 2000 2002 2004 2006
## 1   Brown  297  380  435  391  571  543  408  530  420  326
## 2     RBT  101  142  187  209  245  156  179  321  216  173
```

- There are additional data about the sections of the Blackfoot river for the sampling days each year. Researchers wish to merge these data (shown below) with the data on the fish caught during the sampling period. The `year`, `trip`, and `section` variables are keys that connect the two data sets. How would you merge these two data sets together?

```
head(water)

##   trip year      section temp water_level
## 1    1 1989 Scotty Brown 48.9        3.74
## 2    2 1989     Johnsrud 64.2        3.69
## 3    1 1990 Scotty Brown 53.9        3.37
## 4    2 1990     Johnsrud 65.3        3.69
## 5    1 1991 Scotty Brown 40.1        3.67
## 6    2 1991     Johnsrud 52.0        3.53
```

APPENDIX B

FACULTY INTERVIEW PROTOCOL FROM CHAPTER THREE

1. What computational understandings and/or abilities do you believe are necessary for practitioners in your field?
2. What computational understandings and/or abilities do you expect that masters or doctoral graduates of your program will have?
3. Of the courses required for completion of a masters or doctoral degree, which, if any, specifically teach computing?
4. To the best of your understanding, what computational concepts and/or skills are taught in each of these courses?
5. For the methods that are necessary for coursework and research that are not specifically taught, where do you believe students are acquiring these skills?
6. What resources do you recommend to students who require support in building and applying computational skills?

APPENDIX C

CODEBOOOK OF FACULTY INTERVIEWS

Theme	Description	Example Faculty Quotes
Working with Data (in R)	Students need to perform a variety of tasks for working with data in R	• Getting data into R and working with it: – "I think [students] need to know how to organize their data and get it in in a way that can be used by R." – "I focus on students understanding how to organize the data into spreadsheets, so that you can put them into R." – "I do expect [students] to be able to like work with data in R." • Merging and collating data in R – "I feel like because we collect so many different types of data, I think it's hard for our students to find a way to merge them into one meaningful dataset. You know you can't be analyzing like multiple Excel spreadsheets. " – "Being able to manipulate manage and handle data sets no matter how large or small they may be is an absolute absolutely critical." • Wrangling Data – "You query [your data] to select the subset of the data that you want to analyze at any given moment, you don't open up your spreadsheet with all of your data in it and try and do an analysis."

Theme	Description	Example Faculty Quotes
Working with Data (in R)	Students need to perform a variety of tasks for working with data in R	• Wrangling Data (continued) – "Understanding what the long format is that allows you to do any cross tabulation you want. And being able to choose, being able to figure out how to get the cross tabulation you need, so that you can bring it into R and do your regression, or whatever it is." – "Is typically looks something like [student], where she has lots of different data files coming in and she needs to somehow aggregate across them or merge them in some sort of meaningful way in order for her to implement some type of analysis." – "Yeah, filtering, subsetting, dealing with dates. It's really pretty straightforward stuff and like changing the long to wide form, or vice versa. So those are the big issues." – "A lot of what my students work on are capture or encounter records for animals. So, there's absolutely a need to build encounter histories, sets of 0s and 1s, for when an individual or a species was observed. So, they absolute are doing those kinds of actions to take raw data and make it into something that analysis tool can use."

Theme	Description	Example Faculty Quotes
Working with Data (in `R`)	Students need to perform a variety of tasks for working with data in `R`	• Repeated Actions – "I mean, ideally in R you are using recycling and apply. And loops are there for cases when you can't, like the result of the next iteration is dependent on something that happened in the prior iteration." – "I think there's also and then sort of the more mechanistic stuff is like riding loops and ifelse's and ways of dealing with dealing with data, beyond just manipulating." – "And noticing, like, that you're doing the same process over and over and over and writing a function."
Data Visualization	Students need to create data visualizations for a variety of purposes during their research	• "[Students] need to both visualize and analyze kind of bigger chemical datasets." • "[Students need to] know how to visualize not only like publication ready things, but also how to visualize your data to explore your data to understand it." • "[Students need to] know how to manipulate the data that they can then use to do the analysis, and to make either exploratory plots or plots that they would use in a figure."

Theme	Description	Example Faculty Quotes
Data Visualization (continued)	Students need to create data visualizations for a variety of purposes during their research	• "I don't really think [visualization] is a huge part of the graduate education. Learning those visualization skills, that's what they need. Visualizations and summarizing data, a whole set of programming skills to visualize and explore." • "[Students need to] build a core subset of like how to aggregate information from those data and from there how to visualize those data." • "I think data visualization is a big part of it all, at every stage." • "Being able to visualize the data is probably the next step, even if it's just in an exploratory fashion. Students need to know how to visualize things, exploring patterns in the data set or even just graphing and looking at patterns in the dataset without even getting into really thinking or statistics."
Reproducibility	The importance of students using reproducible, scripted data manipulation	• "Excel is like great for a lot of things, but sometimes when you have lots of data that can just be these errors that are propagated through the spreadsheet. And so, they, I don't know that's been a that's really hard to quality check." • "[Scripted data manipulations] are really important to me as [students] never manipulate their raw data, so you always have that original file. I feel like when they start moving stuff and merging it into just one final is when bad things happen."

Theme	Description	Example Faculty Quotes
Reproducibility (continued)	The importance of students using reproducible, scripted data manipulation	• "I think that it's hard for [students] to understand the importance of like having a file you can go back to that's completely unmanipulated." • "I think if you get through graduate school in a STEM field and you have had no exposure whatsoever of any form of scripting or programming, I think you're handicapped. I mean I don't even care what language it's in. The ability to serialize a process and code it, I think is really critical." • "You know, the general abilities to be in an algorithmic, programmatic thinker in such a way that you're not using Excel to work with your data. Only that you take the raw data and any manipulations that you do are scripted the raw data file is not touched." • "Certainly, to manipulate their data in ways that are repeatable. So, they're not doing it in Excel." • "I encourage [students] to do if not all nearly all their data manipulation in somewhere with a script rather than somewhere in a spreadsheet with mouse clicks and are recorded."

APPENDIX D

PRE-WORKSHOP SURVEY FROM CHAPTER THREE

You are being asked to participate in a research study to understand what methods are the most effective in teaching computational skills in R. Results from this study may provide a better understanding of the computational thinking and abilities of undergraduate and graduate students.

If you agree to participate in this study you will be asked to complete a pre-workshop survey, detailing your demographic information and computational background and a post-workshop assessment of your understanding of the computational techniques covered. Your survey and assessment will be paired and any information that might identify you personally (including your name) will be removed. Only the workshop administrator will have access to your identity.

Your participation in this research is voluntary. You are free to stop participating in the research at any time, or to decline to answer any specific questions. Your participation in this research study is confidential. There are no foreseen risks to participation in this research study.

If you have any questions regarding this research project you can contact me at allisontheobold@montana.edu.

1. I agree to participate in the study. If you agree to participate, you will be asked to complete all of the questions below.

   ☐ Yes

   ☐ No

2. Please enter a unique identifier as follows: Number of pets (as numeric) + First two letters of your last name (lowercase) + First three letters of your current street (lowercase).

3. Please indicate your relevant departmental affiliation. Check all that apply.

   ☐ Agricultural Economics

   ☐ Agricultural Education

   ☐ Animal & Range Sciences

   ☐ Land Resources & Environmental Sciences

   ☐ Microbiology & Immunology

   ☐ Plant Sciences & Plant Pathology

   ☐ Architecture

   ☐ Art

   ☐ Film & Photography

   ☐ Music

   ☐ Education

   ☐ Health & Human Development

   ☐ Center for Biofilm Engineering

- ☐ Chemical & Biological Engineering
- ☐ Civil Engineering
- ☐ Computer Science
- ☐ Electrical & Computer Engineering
- ☐ Mechanical & Industrial Engineering
- ☐ Nursing
- ☐ Agricultural Economics
- ☐ Chemistry & Biochemistry
- ☐ Earth Sciences
- ☐ Ecology
- ☐ Organismal Biology (Botany, Zoology, etc.)
- ☐ Planetary Sciences (Geology, Climatology, Oceanography, etc.)
- ☐ English
- ☐ History & Philosophy
- ☐ Mathematical Sciences
- ☐ Native American Studies
- ☐ Physics
- ☐ Political Science
- ☐ Psychology
- ☐ Sociology and Anthropology
- ☐ Economics or Business
- ☐ Space Sciences
- ☐ Other (please specify)

4. Your current occupation at the university

   - ☐ Seeking Bachelors degree
   - ☐ Seeking Master's degree
   - ☐ Seeking Doctorate degree
   - ☐ Completing a Post-Doc
   - ☐ Faculty member
   - ☐ Staff Member
   - ☐ Other (please specify)

5. How many computer science courses (undergrad or grad) have you taken?

6. What are your previous computer science experiences? List course names.

7. What programming languages do you have experience with? Check all that apply.

   ☐ Python

   ☐ `R`

   ☐ Java or Javascript

   ☐ C or C++

   ☐ Fortran

   ☐ MatLab or Mathematica

   ☐ SQL

   ☐ Other (please specify)

   ☐ None

   ☐ What is a programming language?

8. What are your previous statistics experiences? List course names.

9. What other courses have you taken that require computer programming (e.g. R, GIS, SPSS, STATA, SAS, MatLab, Mathematica, MARK, etc.)? List course names.

10. What operating system is on the computer you are bringing to the workshop?

    ☐ OSX

    ☐ Windows

    ☐ Linux or Ubuntu

    ☐ What is an operating system?

11. Have you participated in independent or collaborative research outside the classroom?

    ☐ Yes

    ☐ No

12. If so, how much? Check all that apply.

    ☐ Little to No

    ☐ A few projects

    ☐ I'm almost done with my thesis

    ☐ I completed a thesis

13. Do you have experience collecting your own data? Check all that apply.

    ☐ Yes, I've helped others collect data.

    ☐ Yes, I've collected my own data,

    ☐ No

14. If you have collected your own data, how did you choose to store it? Check all that apply.

    ☐ Microsoft Excel

    ☐ Microsoft Access

    ☐ Microsoft Word

    ☐ On paper

    ☐ Text file

    ☐ Other (please specify)

15. What is your most important reason for attending this workshop? Check all that apply.

    ☐ Research assistance

    ☐ Coursework assistance

    ☐ Preparation for graduate school

    ☐ Adviser recommended

    ☐ Department/Professor recommended

    ☐ Network with other workshop attendees

    ☐ Refresh or update skills

    ☐ Other (please specify)

16. What resources have you used while learning to program in `R`? Check all that apply.

    ☐ Peers

    ☐ Lab Mates

    ☐ Adviser

    ☐ Course Materials

    ☐ Internet Resources

    ☐ Other (please specify)

17. In a few words, what do you hope to learn from this workshop?

APPENDIX E

POST-WORKSHOP SURVEY FROM CHAPTER THREE

1. Please enter a unique identifier as follows: Number of pets (as numeric) + First two letters of your last name (lowercase) + First three letters of your current street (lowercase).

2. How did you perceive the pace of the workshop?

   ☐ Too slow

   ☐ Slightly slow

   ☐ Just right

   ☐ Slightly fast

   ☐ Too fast

3. How was the balance of lecture to hands-on work?

   ☐ Too much lecture

   ☐ Slightly too much lecture

   ☐ Balanced (lecture/hands-on)

   ☐ Slightly too much hands-on

   ☐ Too much hands-on

4. The Instructor gave clear answers to your questions.

   ☐ Never

   ☐ Rarely

   ☐ Sometimes

   ☐ Often

   ☐ All of the time

   ☐ I never asked a question

5. The Instructor was considerate.

   ☐ Never

   ☐ Rarely

   ☐ Sometimes

   ☐ Often

   ☐ All of the time

6. The Instructor was a good communicator.

   ☐ Never

   ☐ Rarely

   ☐ Sometimes

- ☐ Often
- ☐ All of the time

7. The Instructor was enthusiastic.

   - ☐ Never
   - ☐ Rarely
   - ☐ Sometimes
   - ☐ Often
   - ☐ All of the time

8. There were a sufficient number of helpers available to address any questions I might have.

   - ☐ Yes
   - ☐ No
   - ☐ I never asked any questions

9. The helpers were considerate.

   - ☐ Never
   - ☐ Rarely
   - ☐ Sometimes
   - ☐ Often
   - ☐ All of the time

10. The helpers were good communicators.

    - ☐ Never
    - ☐ Rarely
    - ☐ Sometimes
    - ☐ Often
    - ☐ All of the time

11. The helpers were enthusiastic.

    - ☐ Never
    - ☐ Rarely
    - ☐ Sometimes
    - ☐ Often
    - ☐ All of the time

12. The overall atmosphere of the workshop was welcoming.

    ☐ 1 (Strongly Disagree)
    ☐ 2
    ☐ 3 (Neutral)
    ☐ 4
    ☐ 5 (Strongly Agree)

13. The amount of information covered at the workshop was reasonable for allotted time.

    ☐ 1 (Strongly Disagree)
    ☐ 2
    ☐ 3 (Neutral)
    ☐ 4
    ☐ 5 (Strongly Agree)

14. How much of the information presented at this workshop was new to you?

    ☐ None of it
    ☐ Some of it
    ☐ About half of it
    ☐ Most of it
    ☐ All of it

15. I learned skills that I will be able to use in my research/work.

    ☐ 1 (Strongly Disagree)
    ☐ 2
    ☐ 3 (Neutral)
    ☐ 4
    ☐ 5 (Strongly Agree)

16. How soon do you anticipate using the skills you learn at the workshop?

    ☐ Immediately
    ☐ In the next 30 days
    ☐ In the next 6 months
    ☐ More than 6 months from now
    ☐ I am not sure when

17. I would recommend this workshop to a friend/colleague.

- ☐ 1 (Strongly Disagree)
- ☐ 2
- ☐ 3 (Neutral)
- ☐ 4
- ☐ 5 (Strongly Agree)

18. The workshop was worth my time.

    - ☐ 1 (Strongly Disagree)
    - ☐ 2
    - ☐ 3 (Neutral)
    - ☐ 4
    - ☐ 5 (Strongly Agree)

19. What did you learn from this workshop?

20. Any comments on what was the best thing about this workshop?

21. Any comments on what needed the most improvement at this workshop (e.g. structure, content, location, etc.)?

APPENDIX F

CODEBOOOK OF STUDENT RESEARCH CODE

Codebook of Alicia's Themes

Fall 2018

Theme	Description	Data Science Skills with R Script Examples
Workflow	Structure and organization of performing data analyses in R	• (lack of) reproducbility—Sole comment eluding to data manipulation performed: `#OUTLIER REMOVED` • (lack of) readability – Five comments total in 78 lines of code, fitting six linear models – Lines of code with no indication of purpose or use `qt(.975,9)`

R Environment	Knowledge of the R environment	• temporarily attach objects and select columns from dataframe
		`with(ProximateAnalysisData,`
		`plot(logPSUM ~ Lipid, las = 1))`

Theme	Description	Data Science Skills with R Script Examples

Inefficiency	Operations repeated numerous times, improper use of R script	• Calculations saved in R script which belong in the R console
		`qt(.975,9)`
		• Processes repeated numerous times
		`anterior <- lm(ProximateAnalysisData$PSUA~ ProximateAnalysisData$Lipid)`
		`summary(anterior)`
		`with(ProximateAnalysisData, plot(PSUA ~ Lipid, las = 1)) abline(anterior)`
		`posterior <- lm(ProximateAnalysisData$PSUP ~ ProximateAnalysisData$Lipid)`
		`summary(posterior)`
		`with(ProximateAnalysisData, plot(PSUP ~ Lipid, las = 1)) abline(posterior)`

Theme	Description	Data Science Skills with R Script Examples
Data Structures	Data structures, inherent to the R environment, with which Alicia worked	• dataframe `ProximateAnalysisData$logPSUM`
Data Wrangling	Working with data to prepare for future analyses	• selecting columns `ProximateAnalysisDataOutlier$PSUA`

Data Visualization	Types of data visualizations created, additions and modifications to visualizations	• Scatterplot
		`with(ProximateAnalysisData, plot(Lipid ~ logPSUM, las = 1))`
		• Trend line
		`anterior <- lm(ProximateAnalysisData$PSUA~ ProximateAnalysisData$Lipid)`
		`with(ProximateAnalysisData, plot(PSUA ~ Lipid, las = 1))`
		`abline(anterior)`

Theme	Description	Data Science Skills with R Script Examples

Data Visualization (continued)	Types of data visualizations created, additions and modifications to visualizations	• Customization
		
		`with(ProximateAnalysisData,`
		`plot(PSUM ~ Lipid,`
		`las = 1,`
		`xlab = "Whole-body Lipid Content (%)",`
		`ylab = "UM Fatmeter Reading"))`

Data Model	Statistical methods used to model data	• `lm()`
		
		`middle <- lm(ProximateAnalysisData$PSUM ~`
		`ProximateAnalysisData$Lipid)`

Spring 2019

Theme	Description	Data Science Skills with R Script Examples
Workflow	Structure and organization of performing data analyses in R	• difficulties with reproducbility – data manipulation for original R script carried out in Excel – scripted data wrangling for new R script `early <- subset(RPMA2Growth, StockYear < 2006)` • difficulties with readability – Modified R script contained thoughtful code comments for each linear model, and sections of code delineated with headers – New R script contained only one comment—`#Tanner's code/help`
Data Structures	Data structures, inherent to the R environment, with which Alicia worked	• vectors `RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 1]`

Theme	Description	Data Science Skills with R Script Examples

R Environment	Knowledge of the R environment	• use of R packages
		`library(plyr)`
		• code not reproducible, due to misordering of package loading
		`EarlyWeightAge <- ddply(Early, ~Age, summarise, meanWE=mean(Weight, na.rm = T))`
		`EarlyLengthAge <- ddply(Early, ~Age, summarise, meanLE = mean(ForkLength, na.rm = T))`
		`MidLengthAge <- ddply(Mid, ~Age, summarise, meanLM = mean(ForkLength, na.rm = T))`
		`library(plyr)`

Theme	Description	Data Science Skills with R Script Examples

R Environment (continued)	Knowledge of the R environment	• code not reproducible, due to references to non-existent variables
		
		`LengthAge <- ddply(RPMA2GrowthSub, ~Age,`
		`summarise,`
		`meanL=mean(ForkLength,`
		`na.rm = T))`
		
		`plot(LengthAge$mean ~ LengthAge$Age)`

Data Wrangling	Working with data to prepare for future analyses	• mutating variables
		
		`log(ProximateAnalysisData$PSUA`
		
		• filtering rows
		
		`subset(RPMA2Growth,`
		`StockYear < 2014 & StockYear > 2003)`
		
		`RPMA2GrowthSub$Weight[`
		`RPMA2GrowthSub$Age == 1`
		`]`

Theme	Description	Data Science Skills with R Script Examples

Data Wrangling (continued)	Working with data to prepare for future analyses	• group summaries
		
		`EarlyWeightAge <- ddply(Early, ~Age, summarise, meanWE=mean(Weight, na.rm = T))`
		
		`Weight1 <- mean( RPMA2GrowthSub$Weight[ RPMA2GrowthSub$Age == 1 ], na.rm = TRUE)`

Theme	Description	Data Science Skills with R Script Examples

Inefficiency	Operations repeated numerous times, improper use of R script	• Identical object created twice
		
		`WeightChange <- rep(NA, 9)`
		
		`library(plyr)`
		`WeightAge <- ddply(RPMA2GrowthSub, ~Age,`
		`summarise,`
		`meanW=mean(Weight,`
		`na.rm = T))`
		
		`LengthAge <- ddply(RPMA2GrowthSub, ~Age,`
		`summarise,`
		`meanL=mean(ForkLength,`
		`na.rm = T))`
		
		`#Tanner's code/help`
		
		`WeightChange <- rep(NA, 9)`

Theme	Description	Data Science Skills with R Script Examples

Inefficiency (continued)	Operations repeated numerous times, improper use of R script	• Packages loaded numerous times
		
		`library(plyr)`
		`WeightAge <- ddply(RPMA2GrowthSub, ~Age,`
		`summarise,`
		`meanW=mean(Weight,`
		`na.rm = T))`
		
		`LengthAge <- ddply(RPMA2GrowthSub, ~Age,`
		`summarise,`
		`meanL=mean(ForkLength,`
		`na.rm = T))`
		
		`#Tanner's code/help`
		
		`WeightChange <- rep(NA, 9)`
		
		`library(plyr)`
		
		`WeightAge <- ddply(RPMA2GrowthSub, ~Age,`
		`summarise,`
		`meanW=mean(Weight,`
		`na.rm = T))`

Theme	Description	Data Science Skills with R Script Examples

Inefficiency (continued)	Operations repeated numerous times, improper use of R script	• Processes repeated numerous times (process carried out nine times, changing the value of RPMA2GrowthSub$Age from 1 to 9)
		
		`Weight1 <- mean(`
		`  RPMA2GrowthSub$Weight[`
		`  RPMA2GrowthSub$Age == 1`
		`  ],`
		`  na.rm = TRUE)`
		
		`Length1 <- mean(`
		`  RPMA2GrowthSub$ForkLength[`
		`  RPMA2GrowthSub$Age == 1`
		`  ],`
		`  na.rm  = TRUE)`

Theme	Description	Data Science Skills with R Script Examples

Workflow	Structure and organization of performing data analyses in R	• (lack of) readability
		– 870 lines of code, fitting and visualizing 97 linear models
		– 105 total comments
		– Lines of code with no indication of purpose or use
		`#Favorite Regression`
		`summary(lmKnLipidAllMidKn) #.2389`
		– controlling code by commenting it out
		`# plot(anterior2E)`
		`# plot(posterior2E)`
		`#`
		`# posterior2E`
		`#`
		`# #CI`
		`#`
		`# qt(.975,9)`
		– objects whose names fail to indicate their contents
		`# qt(.975,9)`

Theme	Description	Data Science Skills with R Script Examples

Workflow (continued)	Structure and organization of performing data analyses in R	• (lack of) readability
		– lack of annotations regarding the procedure the script carries out
		`summary(lmPA) #.3958`
		`summary(lmMetricLipidAllMid)`
		`#.2993 (without interaction)`
		`# .4779 with interaction`
		• use of R packages on GitHub
		`install.packages("devtools")`
		`# required to get packages from GitHub`
		`devtools::install_github(`
		`"cardiomoon/ggiraphExtra")`
		`# package from GitHub for interaction plot`
		`library(ggiraphExtra)`
		`# used to make the interaction plot`

Theme	Description	Data Science Skills with R Script Examples

Data Visualization	Types of data visualizations created, additions and modifications to visualizations	• Group colors
		`with(PADataNoOutlier,`
		`plot(Lipid ~ log(PSUA),`
		`las = 1,`
		`col = ifelse(`
		`PADataNoOutlier$`Fork Length` < 260,`
		`"red",`
		`"black")))`
		• Facets
		`densityplot(~Lipid | Length, data = PADataNoOutlier)`
		`bwplot(Lipid~PSUM | Length, data = PADataNoOutlier,`
		`layout = c(1,3))`
		• Density plot
		`densityplot(~Lipid | Length,`
		`data = PADataNoOutlier)`

Theme	Description	Data Science Skills with R Script Examples

Data Visualization (continued)	Types of data visualizations created, additions and modifications to visualizations	• Boxplot
		
		`bwplot(Lipid~PSUM | Length,`
		`  data = PADataNoOutlier,`
		`  layout = c(1,3))`
		
		• legend
		
		`legend("left",`
		`  legend = c("HTHF", "HTMF", "HTLF", "MTHF",`
		`  "MTMF", "MTLF", "LTHF", "LTMF",`
		`  "LTLF", "Swimming", "SDA",`
		`  "Thermal"),`
		`  col = c("red", "red", "red", "blue",`
		`  "blue", "blue", "green", "green",`
		`  "green", "purple", "purple",`
		`  "purple"),`
		`  lty = 1:3,`
		`  cex = 0.8,`
		`  title = "Treatments")`

Theme	Description	Data Science Skills with R Script Examples
Data Wrangling	Working with data to prepare for future analyses	• mutating variables `ifelse(PADataNoOutlier$`Fork Length` < 280, "red", "black")`

Codebook of Ellie's Themes

Fall 2018

Theme	Description	Data Science Skills with R Script Examples

Workflow	Structure and organization of performing data analyses in R	• readability
		– uses code comments to describe the actions the code is performing, and to separate sections of code
		`#### Standard deviation ####`
		`#### Synthetic Error = 0.001 ####`
		
		`# Specify the number of realizations`
		`totalRealizations <- 100`
		
		`# Build an empty data frame to store parameter`
		`# estimates from each realization`
		`respRateEst1 <- numeric(totalRealizations)`
		– uses named arguments to improve readability of code
		`time <- seq(from = 0, to = 23, length = 24)`

Theme	Description	Data Science Skills with R Script Examples
Workflow (continued)	Structure and organization of performing data analyses in R	• reproducbility—every action is scripted, files are organized in a project directory

Efficiency	Methods used to reduce computation time, and prevent operations from being repeated numerous times	• no repeated calculations
		`# Store model predictions for the estimated parameters`
		`confidences3[realization,] <- predict(nlsr)`
		`# Calculate the standard deviation of the residuals`
		`sderr <- sd(synthData - confidences3[realization,])`
		`# Store model predictions with added error to`
		`# generate prediction intervals`
		`predictions3[realization,] <-`
		`  confidences3[realization,] +`
		`  rnorm(length(time), mean = 0, sd = sderr)`
		• use of functions for repeated processes
		`# Load useful functions and packages`
		`source(file = "./hw5_functions.R")`
		`source(file = "./rapper_functions.R")`
		`source(file = "./gangsta_rapper_functions.R")`

Theme	Description	Data Science Skills with R Script Examples

Efficiency (continued)	Methods used to reduce computation time, and prevent operations from being repeated numerous times	• use of for-loop to repeat a process
		`for(realization in 1:totalRealizations){`
		`REPEATED PROCESS:`
		`# Add normally distributed error to`
		`# generate synthetic data`
		`# Estimate parameters by minimizing`
		`# sum of squared residuals`
		`# Get parameter estimate`
		`nlsrest <- summary(nlsr)$coefficients`
		`# Store parameter estimate in numeric vector`
		`# Store model predictions for the estimated parameters`
		`# Calculate the standard deviation of the residuals`
		`# Store model predictions with added error to generate`
		`}`

Theme	Description	Data Science Skills with R Script Examples

Efficiency (continued)	Methods used to reduce computation time, and prevent operations from being repeated numerous times	• use of vectorization to repeat a process
		`predictionInt3 <- apply(`
		`  X = predictions3,`
		`  MARGIN = 2,`
		`  quantile,`
		`  probs=c(0.025, 0.5, 0.975)`
		`)`

Data Model	Statistical methods used to model data	• `nls()`
		`  # Estimate parameters by minimizing`
		`  # sum of squared residuals`
		`  nlsr <- nls(`
		`  formula = synthData ~ USER-DEFINED FUNCTION`
		`  )`

Data Wrangling	Working with data to prepare for future analyses	• mutate variables
		`confidences3[realization,] +`
		`  rnorm(length(time), mean = 0, sd = sderr)`

Theme	Description	Data Science Skills with R Script Examples

Data Wrangling	Working with data to prepare for future analyses	• filter rows
		`confidences3[realization,] <- predict(nlsr)`
		• select columns
		`confidenceResp3["97.5%"] - confidenceResp3["2.5%"]`
		• data summaries
		`quantile(respRateEst3, probs = c(0.025, 0.975))`

Data Structures	Data structures, inherent to the R environment, with which Ellie worked	• vectors
		`respRateEst3 <- numeric(totalRealizations)`
		• matrices
		`confidences1 <- matrix(nrow = totalRealizations, ncol = length(time))`

Theme	Description	Data Science Skills with R Script Examples

Data Visualization	Types of data visualizations created, additions and modifications to visualizations	• scatterplot
		
		`plot(x = c(0.001,0.003,0.005),`
		`y = c(uncert1,uncert2,uncert3),`
		`type = "point",`
		`ylab = "Parameter Estimate Uncertainty",`
		`xlab = "Synthetic Error Standard Deviation",`
		`pch = 19,`
		`ylim = c(min(c(uncert1,uncert2,uncert3)),`
		`max(c(uncert1,uncert2,uncert3))),`
		`col = "darkseagreen2")`
		
		• density plot
		
		`plot(density(respRateEst1),`
		`xlab = "Estimated Maintenance Respiration`
		`rate (kJ per umol biomass per hour)",`
		`las = 1,`
		`main = "Respiration Rate Probability Density`
		`\n Known Respiration Rate = 2.8e-06,`
		`Sum of Squares Objective",`
		`col = "darkorchid4",`
		`xlim = c(2.0e-06, 3.5e-06),`
		`ylim = c(0,7e06)`
		`)`

Theme	Description	Data Science Skills with R Script Examples

Data Visualization (continued)	Types of data visualizations created, additions and modifications to visualizations	• line plot
		`plot(x = c(0, time+1),`
		`y = c(10,knownModelDO),`
		`xlab = "Time (h)",`
		`ylab = "Dissolved Oxygen (umols)",`
		`xlim = c(2,4),`
		`ylim = c(9.2,9.7),`
		`type = "l")`
		• lines
		`abline(v = confidenceResp3["2.5%"],`
		`lty = "dashed", col = "firebrick1")`
		• plotting window
		`par(mai = c(1,1,0.25,0.25),`
		`oma = c(0, 0, 0, 0),`
		`xpd=F)`

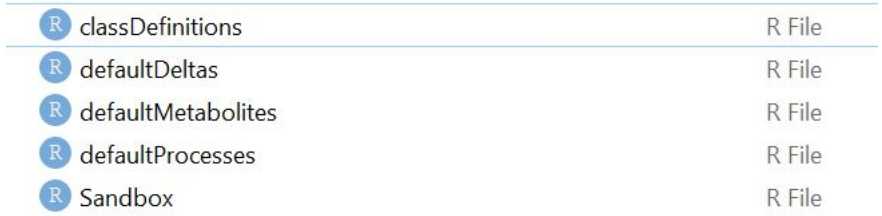
Theme	Description	Data Science Skills with R Script Examples

Data Visualization (continued)	Types of data visualizations created, additions and modifications to visualizations	• colors for groups
		`lines(x = time+1,`
		`y = confidenceInt3["2.5%",],`
		`col = "red",`
		`lty = "dashed")`
		`lines(x = time+1,`
		`y = predictionInt3["2.5%",],`
		`col = "blue",`
		`lty = "dashed")`
		• legend
		`legend("topright",`
		`legend = c("Known Model",`
		`"95% Confidence \n Interval",`
		`"95% Prediction \n Interval"),`
		`col = c("black","red","blue"),`
		`lty = c("solid", "dashed","dashed"))`

Theme	Description	Data Science Skills with R Script Examples

Data Visualization (continued)	Types of data visualizations created, additions and modifications to visualizations	• customization
		
		`plot(density(respRateEst1),`
		`xlab = "Estimated Maintenance Respiration rate`
		`(kJ per umol biomass per hour)",`
		`las = 1,`
		`main = "Respiration Rate Probability Density`
		`\n Known Respiration Rate = 2.8e-06,`
		`Sum of Squares Objective",`
		`col = "darkorchid4",`
		`xlim = c(2.0e-06, 3.5e-06),`
		`ylim = c(0,7e06)`
		`)`

Spring 2019

Theme	Description	Data Science Skills with R Script Examples
Workflow	Structure and organization of performing data analyses in R	• package development—development of R package that allows the user to build custom box-models that simulate the flow of material through any number of pools • file organization—new organization of R script files, specific to working with R6 classes 

Theme	Description	Data Science Skills with R Script Examples

R Environment	Knowledge of the R environment	• R6 Classes
		
		`# Load packages`
		`library(R6)`
		
		`#### State ####`
		`State <- R6Class(`
		`  classname = "State",`
		`  public = list(`
		`  name = NULL,`
		`  value = NULL,`
		`  is.dynamic = NULL,`
		`  initialize = function(name = NULL,`
		`  is.dynamic = F,`
		`  initVal = NULL){`
		`  self$name <- name`
		`  self$is.dynamic <- is.dynamic`
		`  self$value <- initVal`
		`  },`
		`  calculate = function(){`
		`  stop("calculate method does not exist for this class")`
		`  }`
		`  )`
		`)`

Theme	Description	Data Science Skills with R Script Examples

Efficiency	Methods used to reduce computation time, and prevent operations from being repeated numerous times	• vectorization
		`poolStartIsotopicMolVars =`
		`  mapply(function(poolStartMolVar,`
		`  initialIsotopicRatios){`
		`  sapply(initialIsotopicRatios, function(isotopicRatio){`
		`  paste(isotopicRatio, poolStartMolVar)`
		`  }`
		`  )`
		`},`
		`poolStartMolVar = poolStartMolVars,`
		`initialIsotopicRatios = initialIsotopicRatiosByPool`
		`)`

Data Wrangling	Working with data to prepare for future analyses	• mutate variables, using conditional statement
		`if(!(length(elementNames) ==`
		`  length(unique(elementNames)))){`
		`  stop("One or more elementList names are duplicated")`
		`}`

Fall 2019

Theme	Description	Data Science Skills with R Script Examples

Data Structures	Data structures, inherent to the R environment, with which Ellie worked	• dataframes
		
		`# Load data ####`
		`load("/Users/....PATH TO DATA/gas")`
		`load("/Users/....PATH TO DATA/carboys")`
		`gas$days <- as.numeric(gas$minutesSinceAmendment/ (24*60))`

Data Wrangling	Working with data to prepare for future analyses	• select columns
		
		`# Calculate concentration of N15-N14 N2`
		`# relative to Argon`
		`gas$N15_N2_Ar <- (gas$N15_MF * gas$N2Ar)*(40/28.014)`
		`#mol N15-N2 per mol Ar`
		
		• filter rows
		
		`timeD <- (subset(gas, gas$carboy == "D"))$days`

Theme	Description	Data Science Skills with R Script Examples

Data Wrangling (continued)	Working with data to prepare for future analyses	• mutate variables
		
		`# Estimate Initial concentration of N15-NO3`
		`# relative to Ar`
		`N15_NO3_0_D <- 40*((carboys[carboys$CarboyID == "D",]$EstN15NO3) + (0.7*RstN/(1+RstN)))/ (subset(gas, gas$carboy == "D")$Ar[1])`
		
		• character data
		
		`gas[!(substr(gas$sampleID,3,3) %in% c("b","c")), ]`