Teaching Python for Data Science: Collaborative development of a modular & interactive curriculum

Marlena Duda[*,1], **Kelly L. Sovacool**[†,1], **Negar Farzaneh**[1,11], **Vy Kim Nguyen**[1,8], **Sarah E. Haynes**[6], **Hayley Falk**[1], **Katherine L. Furman**[3,4], **Logan A. Walker**[5,1], **Rucheng Diao**[1], **Morgan Oneka**[1], **Audrey C. Drotos**[10], **Alana Woloshin**[7], **Gabrielle A. Dotson**[1], **April Kriebel**[1], **Lucy Meng**[9], **Stephanie N. Thiede**[2], **Zena Lapp**[‡,1], **Brooke N. Wolford**[§,1]

[1]Department of Computational Medicine & Bioinformatics, University of Michigan

[2]Department of Microbiology & Immunology, University of Michigan

[3]Neuroscience Graduate Program, University of Michigan

[4]Michigan Neuroscience Institute, University of Michigan

[5]Biophysics Graduate Program, University of Michigan

[6]Department of Pathology, University of Michigan

[7]School of Information, University of Michigan

[8]Department of Environmental Health Sciences, University of Michigan

[9]Department of Electrical Engineering & Computer Sciences, University of California, Berkeley

[10]Kresge Hearing Research Institute, Department of Otolaryngology–Head and Neck Surgery, University of Michigan

[11]Michigan Center for Integrative Research in Critical Care, University of Michigan

Summary

We are bioinformatics trainees at the University of Michigan who started a local chapter of Girls Who Code to provide a fun and supportive environment for high school women to learn the power of coding. Our goal was to cover basic coding topics and data science concepts through live coding and hands-on practice. However, we could not find a resource that exactly met our needs. Therefore, over the past three years, we have developed a curriculum and instructional format using Jupyter notebooks to effectively teach introductory Python for data science. This method, inspired by The Carpentries organization, uses bite-sized lessons followed by independent

[‡]corresponding author. [§]corresponding author.

[*]first author

[†]first author

Conflicts of Interest

None.

practice time to reinforce coding concepts, and culminates in a data science capstone project using real-world data. We believe our open curriculum is a valuable resource to the wider education community and hope that educators will use and improve our lessons, practice problems, and teaching best practices. Anyone can contribute to our Open Educational Resources on GitHub.

Statement of Need

As women bioinformatics trainees at the University of Michigan (U-M), we experience the gender gap in our field first-hand. During the 1974–1975 academic year, women achieved 18.9% of total Bachelor's degrees in computer and information sciences in the US (National Center for Education Statistics, 2012). By 1983–1984 this peaked at 37.1%, but fell to 17.6% by 2010–2011. We also see this national trend in the training of the next generation of Bioinformaticians at Michigan Medicine. Since accepting its first students in 2001, the U-M Bioinformatics Graduate Program has graduated 66 male and 22 female doctorates as of 2019. This disparity begins at the applicant level; during 2016–2019 the average percentage of females applying directly to the Bioinformatics PhD program was 35.2%, and the average percentage of female applicants listing Bioinformatics as first or second choice in the Program in Biomedical Sciences, U-M's biomedical PhD umbrella program was 41%.

Previous research on women's educational experiences in science, technology, engineering, and mathematics (STEM) have produced various explanations for persistent gender disparities (Benbow & Vivyan, 2016). One explanation is that women often experience stereotype threats that negatively influence their math and science performance and deter them from pursuing STEM as a career (Hill et al., 2010). The majority of our organization's founding graduate students (all women) began coding in our undergraduate careers or later. We wanted to provide a safe environment for local high school women to develop confidence in themselves and their computational skills before college, and be exposed to successful women role models in STEM to counter negative stereotypes.

Girls Who Code, a national organization whose mission is to close the gender gap in technology (Saujani, 2015), was founded in 2012. Because of our personal experiences and the paucity of women in our field (Bonham & Stefan, 2017; National Center for Education Statistics, 2012), we began a Girls Who Code student organization at the University of Michigan in 2017. For the past four academic years we have registered annually as a recognized Girls Who Code Club because the national organization provides name recognition, curriculum resources, guidance for a Capstone Impact Project, and a framework for launching a coding club. Participants in the Club attend weekly meetings at the University of Michigan (when the club is run in person rather than virtually), and are thus largely high school women from the Ann Arbor area. In 2019 we launched our own summer program, the Data Science Summer Experience. When held in person, the Summer Experience is hosted in Detroit to provide the opportunity for high school women outside of Ann Arbor to learn coding skills in an inclusive environment.

The national Girls Who Code organization provides a curriculum that teaches website and application development through programming languages like HTML and Java; however, our biomedical science graduate students generally have limited experience with these

languages and with web development. In contrast, many of us have extensive experience performing data science using the Python programming language. Data Scientist was rated the #1 job in America by Glassdoor in 2016–2019, #3 in 2020, and #2 in 2021 (Stansell, 2019). Furthermore, Python is the most popular programming language according to the PYPL PopularitY of Programming Language Index (*PYPL PopularitY of Programming Language Index*, n.d.). Therefore, we believe career exploration in data science using the Python programming language will optimally prepare our learners for careers that provide financial stability and upward economic mobility. By leveraging the data science expertise of our Club facilitators (hereafter termed instructors), we created a specialized curriculum focused on computational data science in the Python programming language.

Girls Who Code encourages participants to learn programming skills while working on an Impact Project website or application throughout the Club (Girls Who Code HQ, 2021). We created an open source Data Science curriculum that teaches the requisite Python and statistics skills to complete a Capstone Project, where learners explore, analyze, and present a data set of their choosing. Using this curriculum, we employ participatory live coding, where learners type and run code along with the instructor in real time. Using paired activities, our curriculum follows the "I do, we do, you do" didactic paradigm (Fisher & Frey, 2013). We provide open source resources for both in-person and virtual versions of our curriculum, including videos corresponding to each lesson. While we developed this curriculum for our Girls Who Code Club and Summer Experience, we believe that it can be widely used for teaching introductory coding for data science.

Collaborative Curriculum Development

We assembled a team of volunteers involved in our club to develop a custom curriculum to teach introductory Python for data science. We chose the content based on what our learners would need to learn to complete a small data analysis project and communicate their findings to their peers. We divided the content by topic into Jupyter notebooks for each lesson, with each lesson taking approximately 15–20 minutes to teach via live coding. Every lesson has a corresponding practice notebook with additional exercises on the same content taught in the lesson, but using different data or variables. We used a similar development workflow as the U-M Carpentries curriculum (Lapp et al., 2021). Briefly, we hosted the curriculum notebooks in a public GitHub repository to facilitate collaborative development and peer review using pull requests. In the initial curriculum drafting phase, developers were assigned lesson and practice notebooks to write. Once the draft of a lesson was completed, the writer opened a pull request and asked for review from a different developer. The reviewer then provided feedback and approved the pull request to be merged into the main branch after the writer made any requested changes. This way, more than one person viewed each notebook before it could be incorporated into the public curriculum, which reduced mistakes and ensured higher quality content. While teaching from the curriculum at the first Data Science Summer Experience, instructors took notes on their experience and made revisions afterward. Maintainers continue to monitor the repository and resolve issues as they arise.

Following the onset of the COVID-19 pandemic, we quickly pivoted our club to a virtual format. In preparation for the 2020 Summer Experience, we switched to a flipped classroom style following feedback from our club participants that it was too difficult to follow along live coding via Zoom (see Instructional Design).

Curriculum

Our curriculum was designed for high school students with no prior coding experience who are interested in learning Python programming for data science. However, this course material would be useful for anyone interested in teaching or learning basic programming for data analysis.

Learning Objectives

The learning objectives of this curriculum are:

1. Write code in Python with correct syntax and following best practices.

2. Implement fundamental programming concepts when presented with a programmatic problem set.

3. Apply data analysis to real world data to answer scientific questions.

4. Create informative summary statistics and data visualizations in Python.

These skills provide a solid foundation for basic data analysis in Python. Participation in our program exposes learners to the many ways coding and data science can be impactful across many disciplines.

Course Content

Our curriculum design consists of 27 lessons broken up into 5 modules that cover Jupyter notebook setup, Python coding fundamentals, use of essential data science packages including pandas and numpy, basic statistical analyses, and plotting using seaborn and matplotlib (Figure 1) (Harris et al., 2020; Hunter, 2007; Waskom, 2021). Each lesson consists of a lesson notebook and a practice notebook containing similar exercises for the learner to complete on their own following the lesson.

Each lesson builds on those before it, beginning with relevant content reminders from the previous lessons and ending with a concise summary of the skills presented within. As they progress through the curriculum, the learners begin simultaneously working on a data science project using a real world dataset of their choosing. While more time is dedicated to lessons early in the program, the formal curriculum tapers off until the learners are solely applying their skills to the data science project. Through this Capstone Project, learners gain practical experience with each skill as they learn it in the lessons; including importing and cleaning data, data visualization, and basic statistical analyses.

Instructional Design

We modeled our instructional design in the style of Software Carpentry (Wilson, 2016).

1. Each lesson begins with a recapping of the relevant core skills presented in the previous lessons.

2. All lessons are designed to be taught via 15-minute live-coding sessions. This method is used by The Carpentries and is demonstrated to be an effective method that engages learners (Nederbragt et al., 2020; Wilson, 2016) since learners must actively engage with the material and deal with errors and bugs as they arise.

3. Each lesson ends with a summary of core skills presented within the material.

4. Each short lesson is also accompanied by a subsequent 10-minute independent practice, providing further opportunity for practical experience implementing the coding skill at hand and testing learners' understanding of the content.

To better facilitate virtual instruction during the COVID-19 pandemic, we switched to a flipped classroom. Prior to meeting, learners watch videos of instructors explaining the material through "live" coding and code along in the lesson notebook with while watching the video. Each video shows the Jupyter notebook alongside the instructor themselves teaching. Learners then complete a practice notebook corresponding to the lesson. During the virtual meeting time, instructors answer questions and review the core concepts in the practice exercises. This virtual format is especially beneficial because it 1) allows learners to learn at their own pace, and 2) enables dissemination of our curriculum to a wider audience interested in learning introductory Python programming for data science.

For both in-person and virtual instruction, once learners have completed the Fundamentals module and reach the Data Science Essentials module they begin simultaneous work on their data science projects. Projects are completed in a pair programming style, where partners take turns assuming the "driver" (i.e. the typer) and "navigator" (i.e. the helper) roles (Hannay et al., 2009). Switching off in this way helps both partners assume equal responsibility for the project workload, but more importantly it enables improved knowledge transfer through peer-to-peer learning. The culmination of the project is a presentation to peers, instructors, and family members. Through this process learners gain hands-on experience coding, cleaning data, performing statistical analyses, creating informative data visualizations, and communicating their results to others.

In addition to our coding curriculum, another key component of our programming is hosting women guest speakers from diverse fields across academia and industry. Our guest speakers come to discuss the journey they have taken to their career paths as well as how they utilize programming and data science in their jobs. These varied perspectives are extremely valuable to our learners as they provide several practical examples of programming careers in the real world, and expose them to successful women in STEM.

Experience of Use

We have used this curriculum to teach the Data Science Summer Experience and Girls Who Code Club in person in 2019 and virtually in 2020–2021. For both in-person and virtual instances, we had several instructors present at each session to answer questions and help learners debug. Furthermore, one or two instructors were assigned to each project group to help learners define data analysis questions, develop and execute a data analysis plan,

visualize and communicate their findings, and troubleshoot coding problems. Projects have ranged from investigating exoplanets to studying the genomics of psoriasis.

We credit the success of our curriculum not only to the skill of the instructors, but also to the way we organized and executed the lessons and project:

1. The instructors and learners used Google Colaboratory (Colab) to write and execute code in Jupyter notebooks. We chose this option because learners do not have to install any programs to use Google Colab and can easily open and edit the Jupyter notebooks from GitHub. When meeting in person, most learners use Google Chromebooks which have limited programming capabilities, but easy use of a web browser.

2. Assigning instructors to groups allowed learners to build a more personal connection with their instructors, making them feel more comfortable asking questions.

3. Group projects were performed using pair programming to allow learners to collaborate and learn from each other.

4. We used the "sticky note" system from The Carpentries by which learners can ask for help by putting up a colored sticky note (or a Zoom emoji in the case of virtual meetings) (Becker, 2016).

5. We exposed the learners to different aspects of data science by bringing in women guest speakers from academics and industry. This allowed them to better put what they were learning into context, think about how they might use the skills they were learning in potential future careers, and exposed them to successful women in STEM.

Learner experiences—We surveyed learners anonymously after each Club and Summer Experience and found that most felt that their skills in Python programming, problem solving, critical thinking, and collaboration had improved (Figure 2). Furthermore, on a 10 question skills assessment during the 2019–2020 instance of the Club, the average increase in correct answers between the first meeting and the last meeting was 4.2 with a standard deviation of 2.8 (N=5 respondents). We also surveyed Club and Summer Experience alumni and found that 75% (N=20) want to pursue a STEM career. 62% (N=21) are still coding. On a 5-point scale from 'Strongly Disagree' to 'Strongly Agree,' the average answer for 'My participation in GWC impacted my career aspirations' is 4 (s.d.=0.9), with 4.5 (s.d.=0.6) for 'Participating in GWC made me feel more confident in analyzing data' and 3.9 (s.d.=1) for 'Participating in GWC made me more confident in myself.'

Overwhelmingly, learners' favorite parts of the program are the guest speakers and the project. These aspects of our curriculum expose them to new fields and allow them to apply their newfound coding skills to asking an interesting question. A 2021 Club learner shared, "I plan to go to college for Computer Science and get a robotics minor when my college offers it. GWC has inspired me to consider pursuing a Masters or PhD in CS as well as take some electives in Data Science." Five of our 86 alumni have gone on to perform research

with U-M faculty members, with one presenting her work at an international conference. In fact, about a third of participants claim that they are now more interested in pursuing a career in computer or data science compared to before their Girls Who Code experience.

Acknowledgements

References

Becker E (2016). Responding to your Learners. In Data Carpentry. https://datacarpentry.org/blogassessment.

Benbow RJ, & Vivyan E (2016). Gender and Belonging in Undergraduate Computer Science: A Comparative Case Study of Student Experiences in Gateway Courses.

Bonham KS, & Stefan MI (2017). Women are underrepresented in computational biology: An analysis of the scholarly literature in biology, computer science and computational biology. PLOS Computational Biology, 13(10), e1005134. 10.1371/journal.pcbi.1005134 [PubMed: 29023441]

Fisher D, & Frey N (2013). Better Learning Through Structured Teaching: A Framework for the Gradual Release of Responsibility, 2nd Edition.

Girls Who Code HQ. (2021). Girls Who Code Project Gallery. https://hq.girlswhocode.com/project-gallery.

Hannay JE, Dybå T, Arisholm E, & Sjøberg DIK (2009). The effectiveness of pair programming: A meta-analysis. Information and Software Technology, 51(7), 1110–1122. 10.1016/j.infsof.2009.02.001

Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del Río JF, Wiebe M, Peterson P, … Oliphant TE (2020). Array programming with NumPy. Nature, 585(7825), 357–362. 10.1038/s41586-020-2649-2 [PubMed: 32939066]

Hill C, Corbett C, & St. Rose A (2010). Why so few? Women in science, technology, engineering, and mathematics. AAUW. ISBN: 978–1-879922–40-2

Hunter JD (2007). Matplotlib: A 2D graphics environment. Computing in Science Engineering, 9(3), 90–95. 10.1109/MCSE.2007.55

Lapp Z, Sovacool KL, Lesniak NA, King D, Barnier C, Flickinger M, Krüger J, Armour CR, Lapp MM, Tallant J, Diao R, Oneka M, Tomkovich S, Anderson JM, Lucas SK, & Schloss PD (2021). Developing and deploying an integrated workshop curriculum teaching computational skills for reproducible research. bioRxiv. 10.1101/2021.06.15.448091

National Center for Education Statistics. (2012). Digest of Education Statistics. https://nces.ed.gov/programs/digest/d12/tables/dt12_349.asp; National Center for Education Statistics.

Nederbragt A, Harris RM, Hill AP, & Wilson G (2020). Ten quick tips for teaching with participatory live coding. PLOS Computational Biology, 16(9), e1008090. 10.1371/journal.pcbi.1008090 [PubMed: 32911527]

PYPL PopularitY of Programming Language index. (n.d.). https://pypl.github.io/PYPL.html.

Saujani R (2015). Girls Who Code: Annual Report 2015. http://girlswhocode.com/2015report/.

Stansell A (2019). Breaking Down the 50 Best Jobs in America for 2019 - Glassdoor. In Glassdoor Economic Research. https://www.glassdoor.com/research/best-jobs-2019/.

Waskom M (2021). Seaborn: Statistical data visualization. JOSS, 6(60), 3021. 10.21105/joss.03021

Wilson G (2016). Software Carpentry: Lessons learned. F1000Res, 3, 62. 10.12688/f1000research.3-62.v2

Figure 1: Our lesson modules.

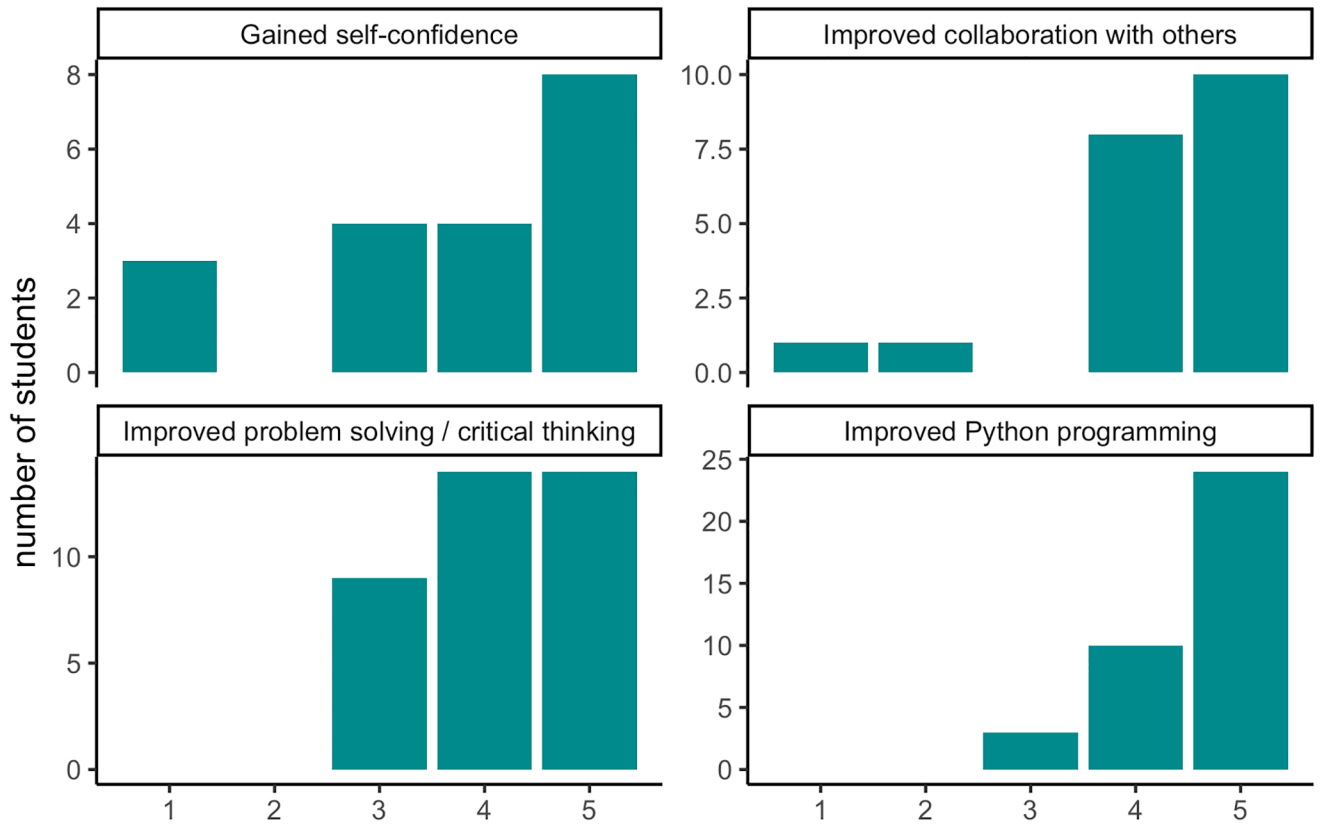All Jupyter notebooks are available on GitHub (https://github.com/GWC-DCMB/curriculum-notebooks).

Figure 2: Post-survey responses.
Learners were asked if they felt that their skills in Python programming, problem solving, critical thinking, and collaboration had improved.