Supplemental Materials

"Cracking the code: An evidence-based approach to teaching Python in an undergraduate earth science setting"

Ethan C. Campbell*§, Katy M. Christensen*, Mikelle Nuwer, Amrita Ahuja, Owen Boram, Junzhe Liu†, Reese Miller, Isabelle Osuna‡, Stephen C. Riser

School of Oceanography, University of Washington, Seattle, Washington 98195

This file includes:

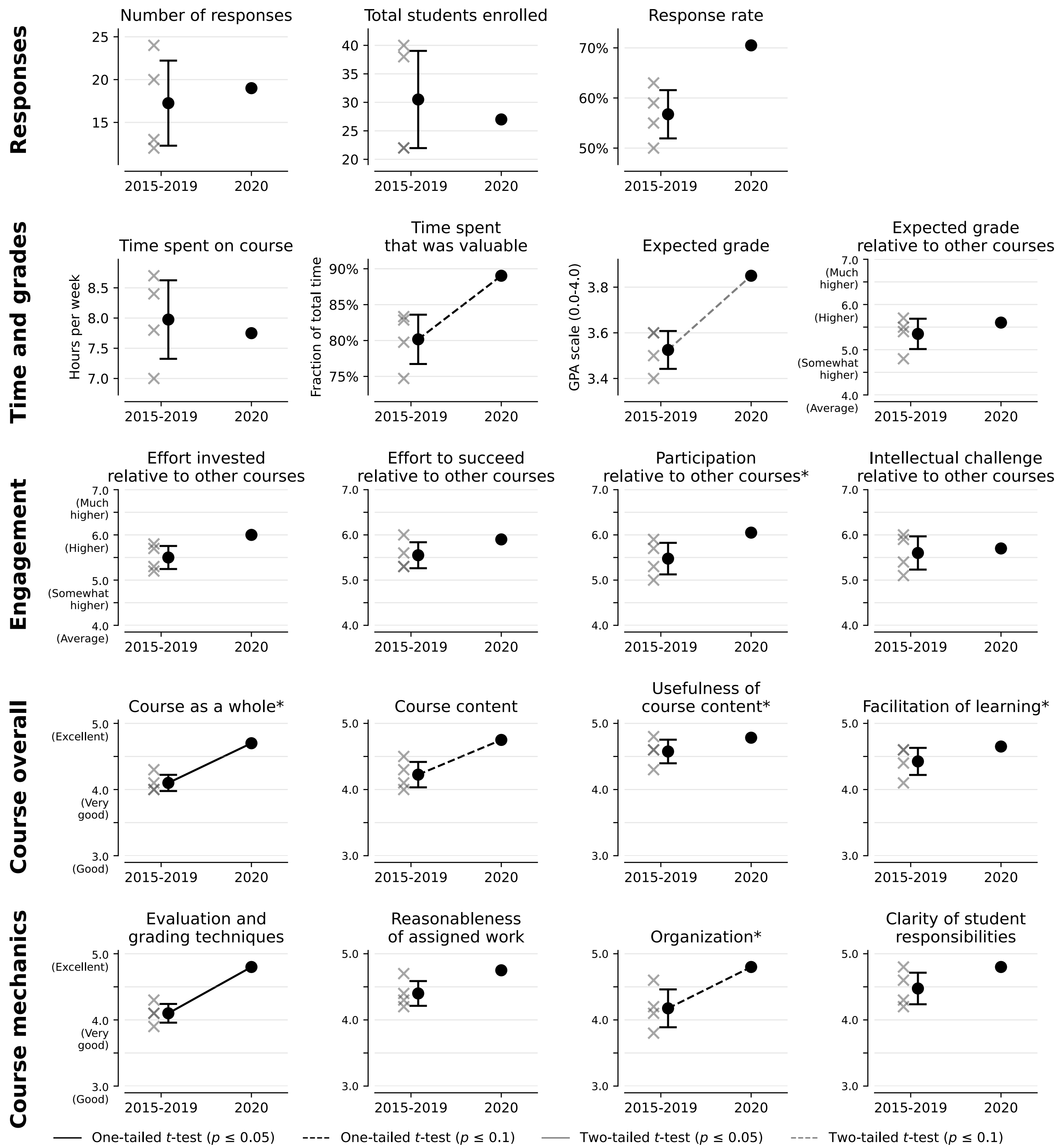
Figures S1-S3
Tables S1-S6
Text S1-S2
Supplemental References

* These authors contributed equally to this work.

§ Corresponding author: Ethan C. Campbell, ethancc@uw.edu, School of Oceanography, University of Washington, Seattle, WA 98195, USA

† Current affiliation: Lamont-Doherty Earth Observatory, Columbia University, Palisades, NY 10964, USA

‡ Current affiliation: School of Earth & Atmospheric Sciences, Georgia Institute of Technology, Atlanta, GA 30332, USA


Responses
Number of responses
Total students enrolled
Response rate
Time and grades
Time spent on course
Hours per week
Time spent that was valuable
Fraction of total time
Expected grade
GPA scale (0.0-4.0)
Expected grade relative to other courses
Engagement
Effort invested relative to other courses
Effort to succeed relative to other courses
Participation relative to other courses*
Intellectual challenge relative to other courses
Course overall
Course as a whole*
Course content
Usefulness of course content*
Facilitation of learning*
Course mechanics
Evaluation and grading techniques
Reasonableness of assigned work
Organization*
Clarity of student responsibilities
2015-2019
2020
7.0 (Much higher)
6.0 (Higher)
5.0 (Somewhat higher)
4.0 (Average)
5.0 (Excellent)
4.0 (Very good)
3.0 (Good)
One-tailed t-test (p ≤ 0.05)
One-tailed t-test (p ≤ 0.1)
Two-tailed t-test (p ≤ 0.05)
Two-tailed t-test (p ≤ 0.1)

[previous page] **Figure S1. All metrics from anonymous end-of-quarter student evaluations in 2015, 2016, 2017, 2019, and 2020** (see Evaluation section "Initial, mid-quarter, and end-of-quarter surveys"). Differently worded questions, indicated with an asterisk (*), were mapped between years as shown in **Table S4** in the Supplemental Materials. Metrics shown are class medians for 2015, 2016, 2017, and 2019 (gray crosses, except for those in the first row ["Responses"]); 2015-2019 mean or 2020 class median (black points); and 2015-2019 standard deviation (bars). Changes from 2015-2019 to 2020 were tested for a significant increase using a one-tailed (black line) *t*-test or a significant change using a two-tailed (gray line) *t*-test at the 95% (solid line) or 90% (dashed line) confidence level according to the methodology detailed in Evaluation section "Initial, mid-quarter, and end-of-quarter surveys." An absence of a line connecting the 2015-2019 and 2020 data indicates no statistically significant improvement or difference. Note that y-axes have been truncated from the full 1-5 scale ("Very poor" to "Excellent") or 1-7 scale ("Much lower" to "Much higher"). Survey questions for which a consistent mapping across years was not possible were excluded; instructor-specific questions are also not shown.

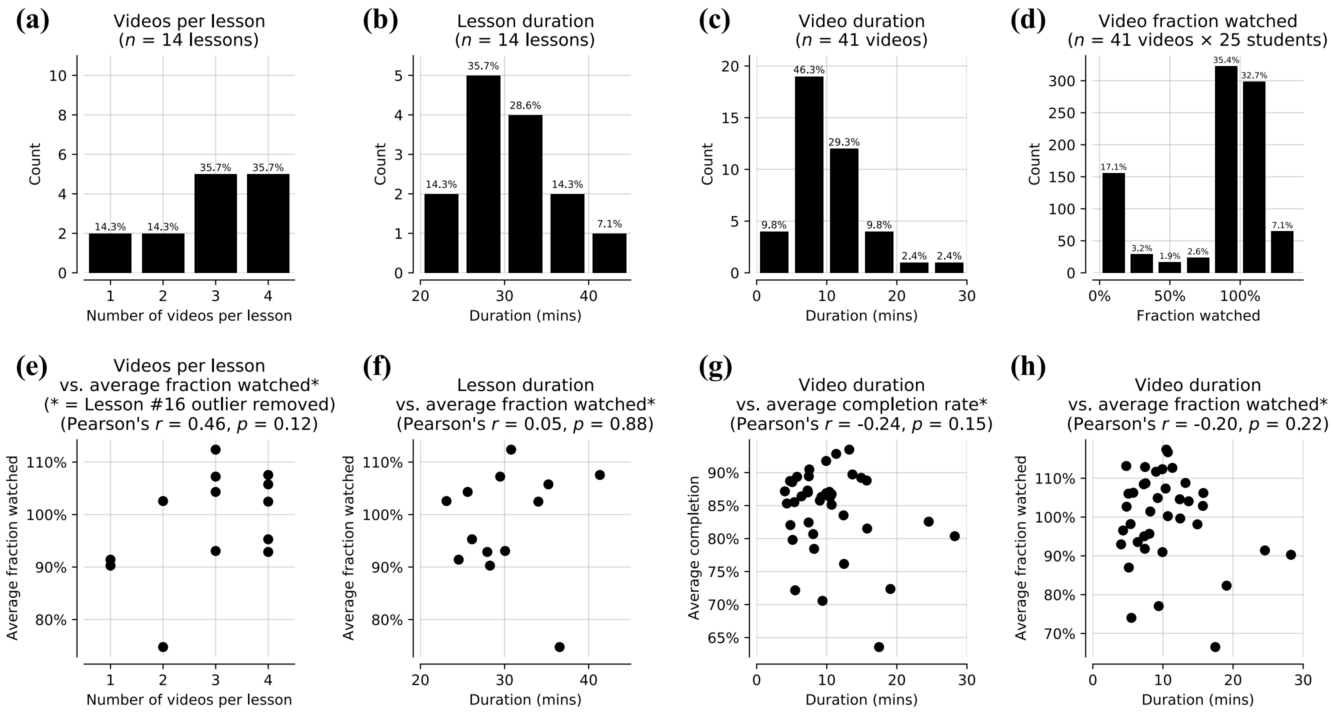


Figure S2. Additional statistics on flipped lesson videos that were posted and viewed on the Panopto platform, based on video-specific metrics obtained from Panopto. Pearson's *r* represents the linear correlation between two variables. Note that none of the correlations tested in panels **(e)-(h)** were significant at the 95% ($p \leq 0.05$) or 90% ($p \leq 0.1$) confidence level. **(a)** Distribution of number of videos included per lesson (as the 14 topical lessons were usually split into multiple videos). **(b)** Distribution of the total duration of lessons. **(c)** Distribution of individual video duration. **(d)** Distribution of fraction of each video watched for each student. Fraction watched represents the total minutes that a specific video was viewed by a specific student divided by its duration, and thus can exceed 100% due to rewinds and repeat views. **(e)** Videos per lesson vs. video fraction watched, averaged across all students. Note that the final video lesson (Lesson #16) was excluded as an outlier due to its lower viewership where indicated using an asterisk (*). **(f)** Lesson duration vs. fraction watched, averaged across all students. **(g)** Video duration vs. completion rate, averaged across all students. Completion rate represents the fraction of a video that was viewed at least once, and thus is capped at 100% for a specific student and video (unlike "fraction watched"). **(h)** Video duration vs. fraction watched, averaged across all students.

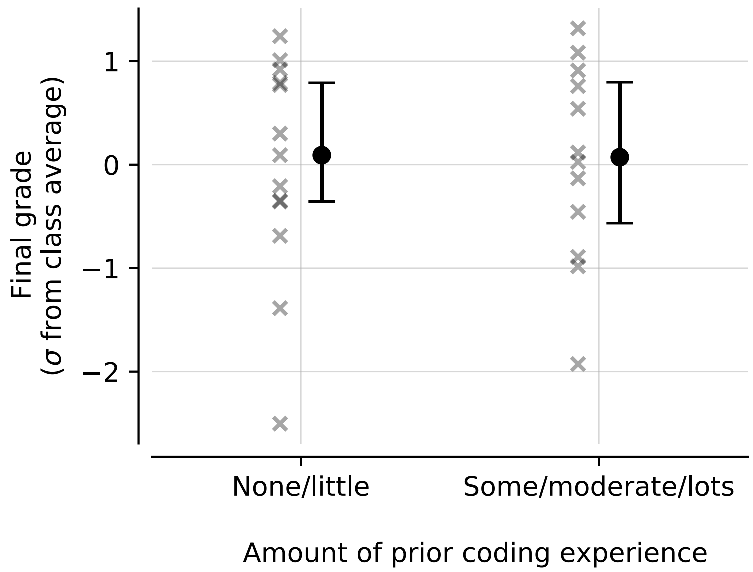


Figure S3. Final course grades dichotomized by amount of prior coding experience. Coding experience was assessed using students' written responses to the Assignment #0 survey (see **Table S3** for rubric and methodology) and is divided here into two groups with none/little experience (score of 1 or 2) and some/moderate/lots of experience (score of 3, 4, or 5) containing approximately equal numbers of students. Final grades were recalculated to ignore two students' incomplete assignments (see Evaluation section "Final grades") and are expressed as standard deviations from the class average (gray crosses). Error bars represent the median and interquartile range (25%-75%) of final grades for each population. No significant difference in final grades was found between the two groups using a two-sided *t*-test ($p = 0.89$).

Table S1. Functions, operators, and methods taught in the course that were used as search terms to assess the complexity of students' final project code. A Python script was used to count instances of each search term in students' project code notebooks, and the number of search terms used at least once (expressed as a percent of all search terms below) is presented as the metric "Python skills used in project" in **Fig. 6**.

Topic	**Search terms**
Basic functions	'len(', 'print(', 'display(', 'range(', 'enumerate(', 'zip(', 'int(', 'float(', 'complex(', 'bool(', 'tuple(', 'type(', 'readline('
Lists	'list(', '.append(', '.extend(', '.insert(', '.remove(', 'del', '.pop(', '.reverse(', '.copy(', '.join(', '.sort('
Strings	'str(', '.lstrip(', '.rstrip(', '.upper(', '.lower(', '.count(', '.replace(', '.split(', '.format('
NumPy	'np.array(', '.dtype', '.astype(', 'np.append(', 'np.insert(', 'np.flip(', 'np.tolist(', '.sum(', '.mean(', '.median(', '.max(', '.min(', 'np.std(', 'np.pi', 'np.e', 'np.inf', 'np.nan', 'np.absolute(', 'np.round(', 'np.sqrt(', 'np.exp(', 'np.sin(', 'np.cos(', 'np.zeros(', 'np.ones(', 'np.full(', 'np.arange(', 'np.linspace(', '.size', '.ndim', '.shape', '.reshape(', ',flatten(', '.transpose(', '.vstack(', '.hstack(', 'np.genfromtxt(', 'np.meshgrid('
Time	'datetime.now()', '.year', '.month', '.day', '.hour', '.minute', '.second', '.microsecond', 'datetime.strptime', 'datetime.strftime', '.total_seconds()', 'timedelta', 'mdates.date2num('
Pandas	'.Series(', '.index', '.values', '.loc[', '.iloc[', 'pd.concat(', 'pd.DataFrame(', '.describe(', '.to_csv(', '.read_csv(', '.read_excel('
Xarray	'.open_dataset(', '.open_mfdataset(', '.attrs', '.isel(', '.sel(', '.item'
SciPy	'stats.linregress(', 'interpolate.interp1d(', 'interpolate.griddata('
Plotting	'.figure(', '.subplots(', '.xlabel(', '.ylabel(', '.set_xlabel(', '.set_ylabel(', '.grid', '.colorbar', '.set_label', '.clabel', '.invert_yaxis', '.gca', '.axes', '.coastlines', '.add_feature(', '.set_extent('
Plot types	'.plot(', '.scatter(', '.hist(', '.contour(', '.contourf(', '.pcolormesh('
Logic	' if ', ' while ', ' for ', ' is ', ' in ', ' not ', ' else:', ' elif ', ' and ', '~', '==', '!=', '>=', '<='

Table S2. Grading rubric for students' final research projects. In the first column, corresponding main student learning objectives (SLOs) are appended to the rubric (see Implementation section "Course history and development" for the full numbered list of SLOs). **Fig. 7** depicts assessments of students' final projects using this rubric, grouped by theme and SLOs.

Corresponding main student learning outcome(s)		Limited (0-50%)	Good (50-75%)	Exceptional (75-100%)	
	Presentation content				
SLO #5 ("Formulate and investigate	Background	Topic background is missing or severely lacking in detail.	Topic background is sufficient, but missing	Topic background is clear, complete, and relevant.	3 points

scientific research questions”)			some details or lacks coherency.		
	Questions / hypotheses	Questions are not well-defined. Hypotheses are not substantiated.	Questions are well-defined. Hypotheses draw on prior knowledge.	Questions are well-defined and pertinent for the topic. Hypotheses draw on prior knowledge and have clear explanations for why they are expected.	2 points
SLO #3 (“Access, read, transform… and interpret oceanographic data with confidence using Python”)	Data collection	Information about the data collection process is missing key details or is inaccurate. The limitations of the data are missing or not realistic.	Information about the data collection process is accurate, but missing some minor details. The limitations of the data are explained.	Information about the data collection process is complete and accurate. Underlying problems and limitations of the data are explained. Use of these data to answer the project questions is justified.	3 points
	Data processing	The student has made errors in processing their data. The student is missing steps.	The student has processed the data correctly. Steps for obtaining, loading, cleaning, and analyzing the data are well-defined.	The student has processed the data correctly and taken precautions to ensure that their results are appropriate. Steps for obtaining, loading, cleaning, and analyzing the data are well-defined.	3 points
	Results / interpretation	Results of the project do not attempt to answer the scientific questions. The data visualizations are not relevant.	Results of the project somewhat answer the scientific questions. Data visualizations are mostly appropriate for the data.	Results of the project answer, or earnestly attempt to answer, the scientific questions. Data visualizations are entirely appropriate for the data.	3 points
	Presentation skills				
N/A	Organization	The presentation is not in a logical order and the student makes no effort to guide the audience.	The presentation is organized in a logical order and takes some care to guide the audience.	The presentation is organized in a logical order and shows exceptional attention to guiding the audience.	2 points
	Timing	The student far exceeds their allotted time and/or has not made an effort to practice.	The student completes the presentation in somewhat over 5 minutes.	The student completes the presentation within 5 minutes and it is clear that they have practiced.	1 point
SLO #5 (“Formulate and investigate scientific research questions”)	Explanation of ideas / information	The ideas and information explained in the presentation were not clear and were not relevant.	The ideas and information explained in the presentation were clear and relevant.	The ideas and information explained in the presentation were exceptionally clear, relevant, and coherent.	3 points

	Code				
SLO #2 (“Write, execute, and debug Python code”), SLO #6 (“Adopt best practices in programming”)	Correctness	The student misuses code and does not produce reasonable results.	The student uses some coding techniques/tools learned throughout the quarter. The analysis produces reasonable answers that can be replicated with some effort.	The student properly and efficiently uses the coding techniques/tools learned throughout the quarter. The analysis produces reasonable answers that can be replicated easily.	8 points
	Functionality	The code does not run and has egregious errors.	The code is mostly able to run, but has some (small) errors.	The code runs efficiently with no errors.	5 points
	Tidiness	The code breaks proper etiquette and should not be shared with others.	The code mostly follows proper coding etiquette. The organization is somewhat lacking and would need review before sharing.	The code follows proper coding etiquette. It is organized and commented effectively so that it can easily be shared with another person.	6 points
	Perseverance	The student has made no effort to work through problems and hurdles.	The student has made some effort to work through problems.	The student has made a gallant effort to work through problems and documented in their code their best understanding of the problems they are facing.	5 points
	Visualizations				
SLO #3 ("… visualize… oceanographic data with confidence using Python”), SLO #6 (“Adopt best practices in… data visualization”)	Plot clarity	The plots are unclear and do not make sense in the context of the project.	The plots are mostly clear and show some thought from the students about ways to present their data.	The plots are extremely clear and are effective tools to help the audience understand the results/analysis.	5 points
	Colormaps	The colormaps are not appropriate for the data being shown.	The colormaps are appropriate for the data being shown.	The colormaps are appropriate for the data being shown and take into account colorblindness, and perceptual accuracy.	3 points
	Proper labels	The plots are missing most/all labels or have improper labels.	The plots are labeled with general accuracy and completion.	The plots are labeled extremely accurately in a way that guides the audience through the figure.	5 points
	Creativity	The student made no effort to create original plots.	The student has made some effort to create original plots.	The student has created original plots that show the data/analysis in an extremely effective manner.	3 points

Table S3. Rubric used to assess students' prior coding experience based on their written responses to the Assignment #0 survey during Week 1 of the course. Students were asked: "Do you have prior coding experience, and if so, with what language?" and "How comfortable do you feel using technology?" Responses to the first question were graded subjectively based on word choice on a scale from 1-5, using the keywords in quotes (e.g., "a little") when present. As noted below, additional points were awarded to weight responses in favor of prior exposure to Python or similar high-level and/or interpreted languages (MATLAB, Java, R). Points were subtracted to account for less relevant prior experience. If no level of coding proficiency was provided, the base number used was from the students' "comfort with technology" statement ("Very comfortable": 4; "Fairly comfortable": 2). Results are used in **Fig. S3** in the Supplemental Materials and presented as the metric "Prior coding experience" in **Fig. 6**.

1	2	3	4	5
No experience	Minimal experience (e.g., "a little", "small", "tiny amount")	"Some" or "moderate" experience	Experience	Experience (with full additions)

Additions (maximum total: +1.0)	Subtractions (maximum total: -0.5)
+0.5 for one of MATLAB, Java, R	-0.5 if response mentions many years since their previous experience
+1.0 for Python or multiple languages	-0.5 if response mentions that their previous experience was not useful

Table S4. Mapping of university-administered IAS final course evaluation questions from 2015-2019 to 2020. The mapping allows the slightly different evaluation questions from the two periods to be compared in **Fig. 2** and **Fig. S1** in the Supplemental Materials. Metrics listed are the median of responses collected for each class.

Paraphrased question	**Original survey question(s) (2015-2019)**	**Original survey question(s) (2020)**	**Metric and units**
Time spent on course	On average, how many hours per week have you spent on this course, including attending classes, doing readings, reviewing notes, writing papers and any other course related work?		Hours per week
Time spent that was valuable	From the total average hours above, how many do you consider were valuable in advancing your education?		Hours per week, expressed as percent relative to response to question above
Expected grade	What grade do you expect in this course?		GPA scale (0.0-4.0)
Expected grade relative to other courses	Do you expect your grade in this course to be:		1-7 scale ("Much lower" to "Much higher")
Effort invested relative to other courses	The amount of effort you put into this course was:		

Effort to succeed relative to other courses	The amount of effort to succeed in this course was:		
Participation relative to other courses	Your involvement in course (doing assignments, attending classes, etc.) was:	Relative to similar courses taught in person, your participation in this course was:	
Intellectual challenge relative to other courses	The intellectual challenge presented was:		
Course as a whole	The course as a whole was:	The remote learning course as a whole was:	0-5 scale ("Very poor" to "Excellent")
Course content	The course content was:		
Usefulness of course content	Relevance and usefulness of course content were:	Average of: "Usefulness of reading assignments in understanding course content was:", "Usefulness of written assignments in understanding course content was:", "Usefulness of online resources in understanding course content was:"	
Facilitation of learning	Amount you learned in the course was:	The effectiveness of this remote course in facilitating my learning was:	
Evaluation and grading techniques	Evaluative and grading techniques (tests, papers, projects, etc.) were:		
Reasonableness of assigned work	Reasonableness of assigned work was:		
Organization	Course organization was:	Organization of materials online was:	
Clarity of student responsibilities	Clarity of student responsibilities and requirements was:		
Instructor's contribution to the course	The instructor's contribution to the course was:		
Effectiveness of instructor's teaching	The instructor's effectiveness in teaching the subject matter was:		
Quality of instructor answers and feedback	Average of: "Explanations by instructor were:", "Instructor's ability to present alternative explanations when needed was:", "Instructor's interest in whether students learned was:", "Answers to student questions were:"	Quality/helpfulness of instructor feedback was:	

Table S5. Open-ended questions asked in university-administered IAS mid-quarter and final course evaluations in 2020. Students' anonymous responses are tabulated in **Fig. 3** and are excerpted throughout this study.

Evaluation period	**Question**
Mid-quarter	What is helping you to learn in this course?
	What is hindering your learning in this course?
	What can your instructor do to improve your learning in this course?
Final	Was this class intellectually stimulating? Did it stretch your thinking? Why or why not?
	What aspects of this class contributed most to your learning?
	What aspects of this class detracted from your learning?
	What suggestions do you have for improving this class generally?
	If this course were offered remotely again, what suggestions do you have to improve the student experience?

Table S6. List of guiding questions offered to undergraduate student focus group for structuring their testimonial submissions, which are presented in **Text S2** in the Supplemental Materials (also see Evaluation section "Student focus group"). Students were encouraged to address one or more of the questions in their submissions.

1. How did your prior experience with coding (or lack of prior experience) impact your experience with the course? If you have prior coding experience and it was self-taught, what do you see as the benefits of learning scientific programming in a structured environment rather than teaching it to yourself? If your prior coding knowledge was learned from course(s), how did we teach programming that was different and more or less effective than those past course(s)?
2. How did the accessibility elements that we implemented (e.g., captioning, syllabus late policy, extensions, not grading on attendance, breaks during class, virtual office hours, making slide decks available, video optional on Zoom, ability to use chat during class, no course prerequisites, extra credit opportunities, etc.) affect your success in the course?
3. How did the expectations and norms established in the course impact your experience?
4. How did you navigate the course policies we created on collaboration and original work? If you worked with a partner on assignments and/or the final project, what was your experience like? Was it productive/challenging/surprising, and how did the technological tools we used (Colab, Zoom) facilitate it? What advice would you give to professors who are teaching a programming course and want to create opportunities for collaboration?
5. How did the key course elements (recorded videos, in-class activities, assignments, final project, etc.) and technological platforms (Google Colab, Piazza, Zoom, Google Drive/Docs, Canvas) help or hinder your learning?
6. Instead of a textbook, we allowed use of external resources (e.g., documentation websites, Stack Overflow, etc.). How did this compare to having a textbook for the course?
7. How did guidance from the instructors and classmates (via Piazza or in class) help you complete assignments and shape and execute your final project?
8. In what ways did the class help you learn about oceanography sub-disciplines (marine geology, chemistry, physics, biology) or other earth science subjects adjacent to oceanography (e.g., cryosphere, meteorology, climate)? What value do you see in teaching programming in an oceanography curriculum rather than a computer science department?
9. How do you feel this course fit into your overall undergraduate education? How did this course prepare you for future research, like your senior thesis? In what ways do you feel more capable now that you have Python in your arsenal?
10. How do you feel this course shaped your career/life goals or motivation to pursue oceanography or data science during and after college?
11. What was it like taking this class during the pandemic? How does this course compare to other classes you've taken remotely during the pandemic?

Text S1. Final project assignment handout. Note that the grading rubric presented as **Table S2** (modified to include corresponding SLOs) was also distributed to students. A list of due dates is not included here.

OCEAN 215 | Autumn 2020 | **Final project**

Project Description

During this course, you will conduct a small scientific research project from start to finish. You will choose a topic, produce a scientific question related to your topic, suggest a hypothesis, locate data that will help support or reject your hypothesis, analyze/visualize this data using Python, and present your findings to the class. Along the way, you will be responsible for giving your input on other students' projects and you will receive input from other students as well. To further reflect the collaborative nature of scientific research, we also encourage you to post any questions or challenges you encounter during this project to the class on Piazza. **If you wish, you may work with a partner on this project. See below for important information if you choose this option.**

The majority of this project will involve writing Python code to analyze and visualize your chosen data. We will dedicate a substantial amount of class time for this work, during which instructors and peers will be present to help you work through coding challenges. Throughout the quarter, there are a number of due dates for different parts (see the table above) designed to guide you through completing your research. The expectations for the deliverables of this project are detailed below:

1. **Topic check-in:** Consider a topic of research that you would like to examine. If you are having trouble identifying a topic, contact the instructors privately on Piazza so they can help you find something that interests you. Once you have identified your topic, create a private note to the instructors on Piazza in the 'final_project' folder that answers the questions below.

   - What research topics or questions are you interested in?
   - What type(s) of data would help you look into those topics/questions?

2. **Data check-in:** Locate data that will help you look into your selected topic. We will set aside some class time for students to work on this. You can start your exploration by using an internet search engine to look up background information on your topic and find possible data sources. You can also use the oceanography data repositories (e.g. PO.DAAC, NASA Giovanni, BCO-DMO, etc.) listed in the Class #1 slides. As always, the instructors are also available to help you locate a fitting data source. In the 'final_project' folder of Piazza, respond to the data check-in post with answers to the questions below. Make sure that your response is visible to the whole class.

   - What data set(s) do you plan to use?
   - What is one scientific question that you might be able to answer using these data?
   - What is your hypothesis? What do you anticipate the answer to your scientific question is, and why? (try to bring in scientific knowledge from previous courses, published literature, and/or reliable internet sources)

3. **Piazza responses:** Respond to at least 3 other data check-in posts written by your classmates on Piazza with an additional question that they might be able to investigate using their data or about their topic. To reflect the collaborative nature of research, where colleagues often help to dictate research priorities, you will choose one question suggested by a classmate and one question of your own to investigate.

4. **Project presentations:** Present the results of your project to the class in a 5 minute presentation. Presentation schedules will be posted to Canvas later this quarter. Your presentation should include the following:

   - Scientific background on your topic [~1 slide]
   - Two scientific questions (yours and a classmate's from Piazza) with your hypotheses [~1 slide]
   - Information about your data (How/when/where was it collected? What instruments were used? Are there any limitations to your data?) [~1 slide]
   - Your process for obtaining, loading, cleaning, visualizing and analyzing the data. Describe your data file format(s) and structure(s) as well as any challenges you encountered [~1-2 slides]
   - Answers to your scientific questions with associated plots and an explanation of your analysis results [~2-3 slides, ~2-3 figures]

5. **Slides and code:** Submit the slides from your project presentation, your data files/folders, and the code you wrote to analyze your data and create your figures. Your code should follow proper coding etiquette and your figures should be formatted properly. To submit your code, data, and slides, save them and put them in your individual class Google Drive folder. There is no written essay required for this project.

NOTE: Piazza posts that are required for the final project do not count towards the required 5 Piazza posts detailed in the syllabus.

Pair programming option

If you wish, you may work with a partner on this project. This could be a valuable opportunity to experience a research collaboration, work through coding challenges together, and accomplish even more analysis! If you choose this option, the following expectations supersede (override) the requirements listed elsewhere in this document:

- Starting with the **data check-in**, you and your partner may choose a single data set together, and share this identical data set on Piazza. However, please each offer a **different scientific question and hypothesis** in your Piazza posts (i.e. the two of you will come up with two questions and two hypotheses in total).

- For the **Piazza responses**, you and your partner should **each respond to three classmates' posts** (not including your partner's post), for a total of six posts. You will jointly choose **one question from a classmate and two questions of your own** to investigate for your project, for a total of three questions to investigate.

- For the **final project presentations**, please prepare a single **8-10 minute slideshow**, instead of a 5 minute slideshow. Include at least the number of slides specified above for each category. Trade off roles when presenting (i.e. each person should be presenting for about 4-5 minutes).

- You may submit separate Colab code notebooks, or a single joint Colab notebook. However, in all notebooks submitted, please indicate which student wrote each section of code using Python comments. **We expect that both partners will contribute approximately equally to writing code for the project.**

- You will be graded jointly and **will receive the same grade for the project**, except under extenuating circumstances to be determined on a case-by-case basis.

Grading breakdown

Project Part	Grading	Points
Topic check-in:	Complete/Incomplete	10 points
Data check-in:	Complete/Incomplete	15 points
Piazza responses:	Complete/Incomplete	15 points
Project presentation:	See rubric	20 points
Code:	See rubric	40 points
		(100 points total)

Text S2. Testimonials shared by undergraduate student coauthors (see Evaluation section "Student focus group" for more details). The students were encouraged to address one or more of the guiding questions listed in **Table S6** in the Supplemental Materials in their submissions.

———

Other coding classes that I have taken have generally failed to place skills in the context of applications. Without examples of methods being used, there is less of an incentive to understand them. In contrast, this course provided the opportunity to work with oceanographic data, allowing us to recognize the significance of the methods we were applying. For instance, ocean glider data was used to teach about interpolation. This was engaging because we first visualized the original, non-interpolated data and could see the gaps due to the physical motion of the device, then compared this with the data interpolated using the same axes and color scale.

Additionally, the lack of a textbook in this course made it easier to approach methods beyond what we learned in class. Instead, we learned to answer questions by accessing online resources like Stack Overflow. Doing so developed essential skills and gave me the confidence to apply new concepts in my final project. This meant my research could be dictated by my curiosity and questions, as it should be, and not by the limitations of what concepts we had covered in class.

In general, research can seem intimidating to many students because it relies on an individual's creativity. In other classes with exclusively rigid assignments and predetermined tasks, there is little opportunity for students to form original ideas, let alone develop them. In this class, we used creativity and critical thinking skills to develop a final project that answered an independently formed question. This experience has helped to prepare me for research. -O.B.

———

I previously took a Fortran class at the Ocean University of China, which had two traditional lectures and one lab each week. In that class, most students were not engaged during the lectures, which led them to be bewildered when doing real coding. I have also been teaching myself MATLAB for three years, basically learning by doing tasks with the help of the internet. This process has often been time-consuming, and it has been hard to organize my notes in a logical way. In comparison to those experiences, this course provided a logical pathway into Python, especially for oceanography applications. Without this class, it would have taken ten times longer to acquire the same knowledge, which would also have been less clear.

In class, Zoom breakout rooms forced everyone to discuss and practice the coding, which in turn forced us to come well-prepared for class. Though Google Colab has limited storage (RAM) and is unable to process large data sets, it is great for starters. Most of my other classes have been about theory and previously derived conclusions in the field, but this class has provided a bridge between theory and practice. After taking this course, I would say that we can now start to connect math and data to discover the areas of science we are interested in. -J.L.

———

I have always viewed research as something that is extraordinarily complicated. This class demonstrated that knowing a few basic Python functions and packages can provide a solid foundation to start conducting research. Additionally, offering this class as part of an oceanography curriculum instead of relying on a computer science department allowed us to learn about programming skills in a way that directly applied to our interests and studies.

I liked the way that the course was set up, in which we learned the material in an asynchronous video first and then practiced it in class. This helped me to discover where my gaps in understanding were and to learn from other people who may have understood a concept better than I did. Google Colab may not be the most powerful programming platform, but it is streamlined and easy to use, which made it great for first-time coders like me. Piazza was also an exceptionally useful resource.

Many classes present an idealized version of how research works. This class didn't. It was an important learning experience when my final research project didn't yield the correlation I expected. This was frustrating since I put so much time and effort into the project, but it showed that a lack of correlation can be an important result and that one's research doesn't always have to produce a major scientific breakthrough. -R.M.

———

I came in with a little prior coding experience thanks to robotic projects that I completed with my father as a child. In taking this class, the love of coding that I had as a child was reignited. I hadn't realized how beneficial and necessary knowing a programming language would be for research. Having Python in my arsenal opened up research opportunities that I wouldn't have been qualified for before and can aid me in branching out beyond oceanography in the future. The great experience I had in this class – and my realization that research and coding are extremely integrated – inspired me to pursue a minor in Data Science.

In this class, the coding assignments were based on real-world problem solving. I loved having the opportunity to work with a partner because we coded in completely different ways, and it was fascinating to see those differences. We were more effective together because we learned to compromise and collaborate to find the cleanest and fastest method between the two of us. Writing code on Zoom was a good alternative to in-person collaboration because we could share our screens and help pinpoint issues in each other's code. In addition, Piazza was helpful for me because it allowed anonymous or private questions, which avoids the uncomfortable feeling of asking a question that you think might be silly. I liked that we were able to get quick and helpful feedback on our code. It was a better way of communicating than those I have used in other classes, like email, which might get drowned out in a teacher's inbox, or Slack, which doesn't provide the anonymity that Piazza does. -I.O.

Supplemental References

Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory Into Practice*, *41*(4), 212–219.