International Journal *of* Designs for Learning
2021 | Volume 12, Issue 3 | Pages 102-117

DATAWHYS PHASE 1: PROBLEM SOLVING TO FACILITATE DATA SCIENCE & STEM LEARNING AMONG SUMMER INTERNS

Linda Payne, Andrew Tawfik, & Andrew M. Olney, *University of Memphis*

This design case details a data science summer learning experience designed by University of Memphis faculty for HBCU students (NSF #: 1918751) with recruiting assistance provided by LeMoyne-Owen College. The summer learning experience included elements of didactic and collaborative problem-solving during the first five weeks of the internship, followed by a three-week, team-based, problem-solving project using real-world data. While the course was originally designed as a face-to-face learning experience, the impact of COVID-19 necessitated a shift toward online digital spaces. The design case details the opportunities and challenges of STEM online learning and especially underscores the limitations of (a) existing data science technologies for instruction, (b) the shift toward instructional design of materials that supported more self-directed learning, and (c) collaborative problem-solving. Implications for design and practice are also considered.

Linda Payne is a Research Assistant in the Instructional Design & Technology Studio at the University of Memphis. Her research interests include human-computer interaction, problem-based learning, trauma-informed education, content design, informal learning, and computer-supported collaborative learning.

Andrew Tawfik is an Assistant Professor of Instructional Design & Technology at the University of Memphis where he also serves as the director for the Instructional Design & Technology Studio. His research interests include problem-based learning, case-based reasoning, case library instructional design, and computer-supported collaborative learning.Andrew M. Olney, PhD., serves as Professor in both the Institute for Intelligent Systems and the Department of Psychology at the University of Memphis. His research interests are in natural language interfaces, vector space models, dialogue systems, unsupervised grammar induction, robotics, and intelligent tutoring systems.

https://doi.org/10.14434/ijdl.v12i3.31555

INTRODUCTION

Theorists and economists have noted the importance of Science, Technology, Engineering, and Mathematics (STEM) education equity and its impact on economic growth development, national security, and global competitiveness (Stehle & Peters-Burton, 2019). One of the STEM subdomains includes data science, which uses elements of computer science, machine learning, and statistics to solve 'big data' problems. Although STEM and data science are described as important, research shows that underrepresented students have less access to STEM educational programs (Smith, Trygstad & Banilower, 2016). Along those same lines, there has been considerable effort toward equitable learning opportunities within historically black colleges and universities (HBCUs) (Ladson-Billings, 2006; Morton, 2020; Palmer et al., 2011).

To better engender STEM expertise and address some of the systemic disparities within education, additional efforts have focused on learning opportunities outside of traditional classrooms (Simpson & Maltese, 2017). To date, these have often been done in after school programs, libraries, and other informal learning contexts. The learning experience described in the design case used an internship approach whereby students would be mentored by faculty, which allowed us to leverage some of the benefits described in the cognitive apprenticeship literature. Specifically, this design case details how the University of Memphis conducted a summer internship program designed to educate interdisciplinary learners from HBCUs about data science. This program is part of a larger DataWhys project (NSF #: 1918751) that will attempt to teach data science across different populations and levels of expertise. However, a challenge emerged in that the program rapidly shifted toward an online format due to the pandemic brought about by COVID-19. The design case thus details interesting insights about agile instructional design, as well as supporting STEM learning using technology for diverse populations.

INTERNSHIP DESIGN OVERVIEW

The two goals of the DataWhys project are to better understand how people learn data science and, using this understanding, to create optimal supports for learning data science. Because data science is generally viewed as having deep prerequisites in programming, machine learning, and statistics, our research plan includes both cross-sectional studies with participants of different skill levels, as well as longitudinal studies that can closely track a group of participants as they learn over time. Therefore, the summer internship is both a research activity aligned with the larger goals of the grant, as well as an outreach activity designed to increase Black representation in data science at a local and national level. Unlike a formal study where there is a demarcation between researchers and participants, our interns are research partners who provide ongoing feedback on our training materials and methods while they share their experiences of encountering data science for the first time.

The plan for the first phase of the DataWhys project was to create a summer data science learning program for college interns enrolled in a local HBCU. Prior to this internship, we theorized that students could learn how to program with visual blocks and learn data science concurrently, as opposed to learning how to program first, and then learning data science. To test this theory, we married the two learning programs using a computational notebook, called Jupyter Notebook, embedded with data science instruction, along with a software plug-in for teaching programming using visual blocks, called Blockly. A computational notebook is a virtual notebook that combines aspects of word processing software with a programming environment, in this case, Python. Computational notebooks are widely used by professional data scientists because they combine reports with executable code, supporting sharing, replication of results, and extension of analyses.

The learning experience for interns was originally intended to be an in-person learning program whereby students would receive face-to-face instruction and then work within the lab spaces with peers and faculty at the University of Memphis. Our face-to-face design was informed by best practices in NSF Research Experiences for Undergraduates (REU) programs that broaden participation in computing (Morreale et al., 2011), in addition to a well-designed summer internship in Cognitive Science at Stanford (CSLI). However, the design of these programs presupposes that the students involved are majoring in a related discipline, which in the data science case would be computer science, machine learning, or statistics. This is also true for the Data Science for Social Good fellowship at Carnegie Mellon University, which we discovered after we had launched our internship. A prior experience requirement is inconsistent with the overall goal of our project, which is to broaden participation in data science. Therefore, we planned to have a data science "boot camp" for the first part of the internship, followed by a project phase that would be more typical for an REU, with interns working one-on-one with faculty mentors. Because the boot camp would be a shared experience, we were discussing the tradeoffs between bringing the interns into a single room vs. having them stay in separate labs with faculty/graduate student mentors for this phase when COVID-19 shut down our campus in March of 2020.

The advent of COVID-19 caused the design team to rethink the learning experience for these students from in-person to completely online, which had implications for teacher-student interaction, scaffolding strategies, breadth vs. depth of a topic, and learner engagement. Because the future implications of the pandemic were unknown, we knew that some aspects of the online instruction would need to be built "on the fly." Therefore, we would need a more agile instructional design that integrated proven design principles within the online technical tools as they were tested, selected, tweaked, and ultimately incorporated into the learning program. This also meant some trial and error on the part of instructors and designers, as we learned from the practical implementations, and feedback from group reflections with students. That said, we knew that we would use what we learned from this first phase of the project to rework, refine, and redesign the instructional program as appropriate to improve learning. As we discuss in this design case, some specific design tensions and decisions included the following:

a. existing data science technologies for instruction,
b. the shift toward instructional design of materials that supported more self-directed learning, and
c. collaborative problem-solving

The summer internship course consisted of multiple parts throughout each day (Monday—Thursday). In the morning, learners were provided varying degrees of didactic instruction on different data science topics from University of Memphis faculty members. Learners then used Jupyter Notebooks with a Blockly plug-in to engage in individual and collaborative problem-solving for the specific topic. As the instruction shifted toward the afternoon, learners solved a novel task to support learning transfer. The first five weeks were spent building foundational skills instruction; and the last three weeks were focused on team projects, where students applied their skills to complex, real-world problems.

STAKEHOLDERS

Description of Learners

A unique aspect of the learning problem is that it is designed to attract learners from a range of domains and approach the importance of data science from an interdisciplinary perspective. In the context of this design case, the summer interns were HBCU undergraduate students whose career interests ranged from medicine to computer science to

business; however, they all had some interest in learning about data science. The summer internship included seven undergraduate students with varying levels of knowledge and experience with data science. Although this widened the potential student perspective, this also played a role in terms of prior knowledge, specific scaffolding needs, and uneven background knowledge across teams.

As anticipated, students with computer science backgrounds met many of the basic technical aspects of the learning program with more ease, but even they were challenged by programming and needed to use the blocks to program for quite some time, especially in a full online context. That being said, student interest level and conscientiousness seemed to play a significant role in the level of engagement, as evidenced by the level of participation in both the exercises and discussions among the students, regardless of whether or not they had a background in computer science. A concerted effort was made on the part of instructors to create teams of students with various levels of experience to encourage informal learning among the group.

Description of Instructors

On the instructor side, the interdisciplinary project team included University of Memphis faculty members and researchers, as well as graduate students. Faculty members were from Artificial Intelligence, Computer Science, Instructional Design, Psychology, and Statistics departments. The instructors played multiple roles on the project. For the specific instructional times, they provided approximately 15 minutes of didactic instruction (e.g., morning lecture) and facilitated collaborative problem-solving in the morning. They also were available throughout the day to field questions and resolve any technical issues. Each afternoon following student work on problem-based exercises and peer review, instructors and graduate assistants facilitated online sessions with the team of students to reflect upon the learning for that day. In addition, different instructors would have a weekly lunch discussion to share their personal backgrounds and professional development. In separate sessions, instructors offered professional development sessions including professional ethics, giving a good presentation, and applying to graduate school. To further support them, students were also given the opportunity to connect with mentors following the internship.

Description of Designers

Faculty members at the University of Memphis worked together to design the summer learning program for the interns, which, as stated earlier, was part of a larger project called DataWhys. The design decisions for the summer internship included the following: (a) determining the scope of the instructional content, (b) finding the right software for online instruction, (c) adding plug-ins and otherwise manipulating that software to suit learning needs, and (d) finding supportive technologies to promote communication among students and faculty. The lead principal investigator had support from a team of five other principal investigators on the project. Graduate assistants in the field of instructional design were also on hand to help with research and instructional techniques, providing a more student-based perspective.

SUMMER INTERNSHIP INSTRUCTIONAL ACTIVITIES—PART 1

The summer internship was an eight-week course, where the first five weeks were designed to teach learners about various data science topics, including exploratory data analysis, k-nearest neighbors, decision trees, and other related concepts. Because of the nature of the content (computer/data science), the designers focused on creating a problem-based curriculum that included worked examples with decreased amounts of assistance, or fading, as instruction progressed. To start each day, participants would navigate to a landing page where they would review an outline for the day. Each day of instruction consisted of a variety of activities, including an initial standing meeting (didactic instruction), a morning assignment (worked examples), an afternoon session (isomorphic problems), a peer review session, and a reflection hour. The following section provides more detail into each of these areas.

Initial Standing Meeting

Each morning started with a 15-minute standing meeting led by that morning's instructor. Since topics changed daily and lessons were taught by different teachers, the morning meeting was established to recap the previous day's lesson, as well as to set expectations for the current day. The morning session allowed the team and students to discuss a brief lesson on the topic of instruction, the schedule, and the names of the teachers who would be leading that day's classes. This was also a time for students to ask any general questions prior to starting their student-centered learning. Lessons often built upon one another and the program encouraged self-directed problem-solving, so instructors wanted an approach that would help ensure students understood the previous lesson before starting a new one

Jupyter Notebook Morning Assignments

Because the internship approach was designed to develop students' competencies with data science concepts and tools, we wanted them to use tools that practitioners would use. For daily work, we chose Jupyter Notebook, which is an open-source application that allows computer/data scientists to work within computational notebooks to develop code. To avoid installation and technical requirements for the interns, who were using their own computers, we hosted a Jupyter Hub, which is a server-based instance of Jupyter

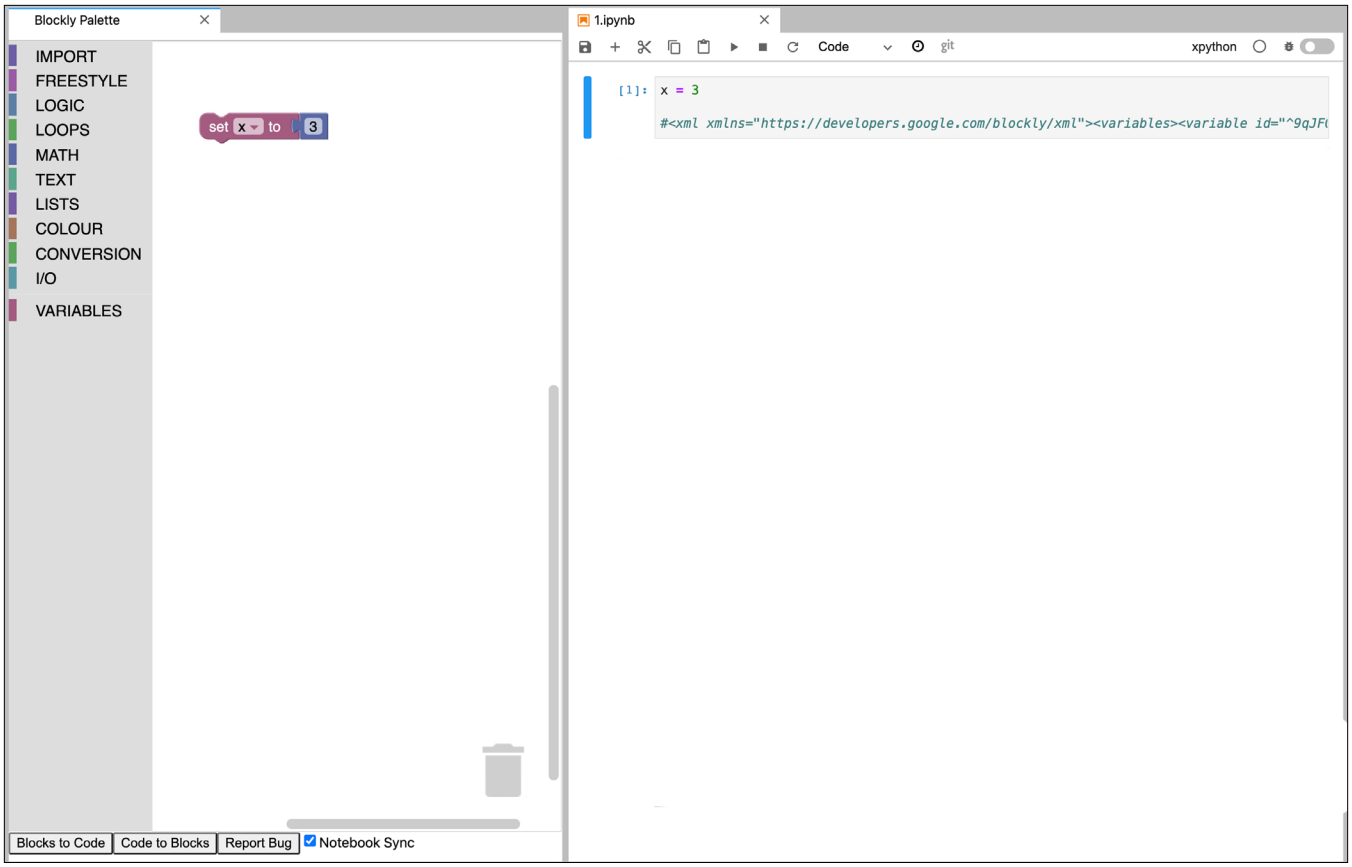


FIGURE 1. Jupyter Notebook with Blockly Plug-In.

Notebook that only requires a web browser to use. In each notebook, users could run code and embed other media relevant to the data science tasks required for the learning objectives of the course.

Interns used Jupyter Notebook with an extension, or plug-in, for Blockly, a library of visual blocks used for programming and editing code. The Blockly extension contained all the blocks interns would need to solve data science problems, and by pressing a "Blocks to Code" button, they could insert the corresponding Python code into their Jupyter notebook. Given the learners in this instructional context were novices, this format thus created a scaffolded environment to solve data science problems in class. Those who needed the assistance and scaffolded nature of blocks could use them, and those who felt more comfortable writing code could choose not to use the blocks. Figure 1 shows an example of blocks arranged in Blockly (left) that have been converted to Python code in a Jupyter notebook (right, blue bar) and then run to produce a table output.

The notebooks were a unique aspect of the learning experience for multiple reasons. One of the design tensions we carefully considered was providing the right amount of depth of the content, as well as the appropriate resources as students directed their problem-solving. We also wanted to respond and scaffold appropriately to novices' emergent needs, especially in a fully online format. At the outset of the project, we compiled a list of nearly 1,000 resources that students could reference; however, this became quickly challenging for multiple reasons. First, each resource in the list contained important information toward the learning objectives, but it was difficult to know which excerpts were relevant given the learning objectives and scope of the Datawhys program. Given the intensity of the internship program, we wanted to limit extensive time for open-inquiry of novel information and instead focus on application of the data science principles. Although the information may have been helpful across the resources, students would invariably encounter duplicate information and thus result in cognitive overload as they evaluated various texts. There was also an issue of cost and copyright for the different texts.

As an alternative, the design team decided to utilize Jupyter Notebook as both a workspace for hands-on activities and a data science resource. In the first portion of the online notebook, we embedded instruction for the given data science topic. Like many textbooks, this provided an overview of the topic, defined major terms, and presented a rationale for why the concept is important to practice. We felt as though this design was particularly important given the online approach. In the latter part of each notebook, students were given an opportunity to test out various aspects of what

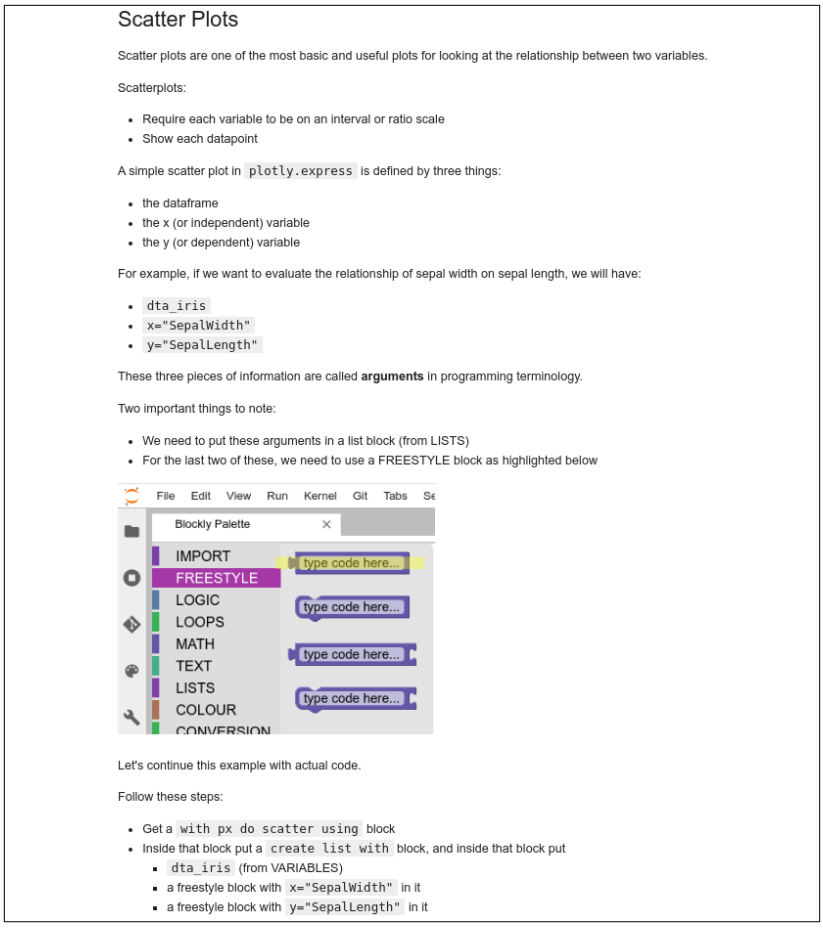


FIGURE 2. Jupyter Notebook Lesson on Scatter Plots—Part 1.

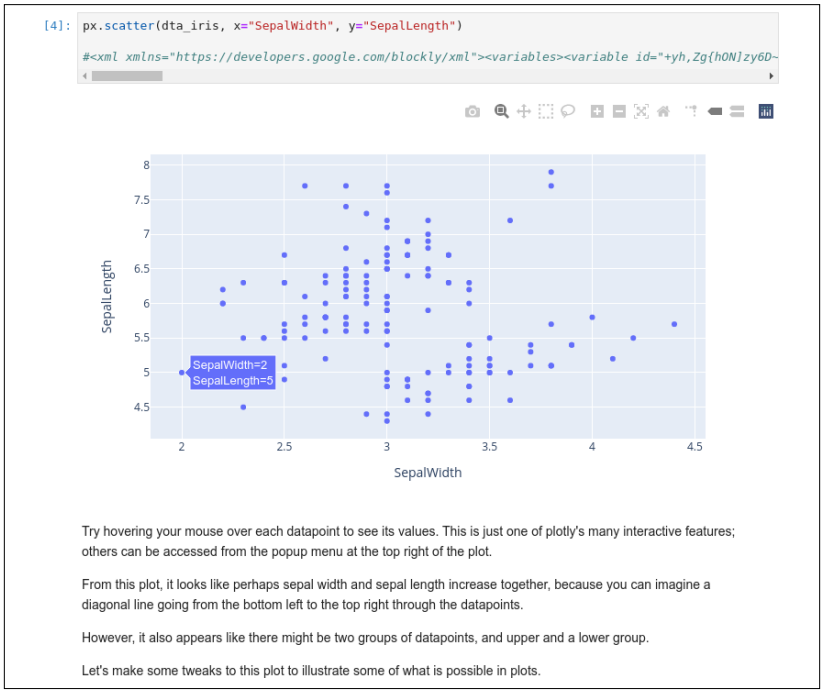


FIGURE 3. Jupyter Notebook Lesson on Scatter Plots—Part 2.

they had read within that notebook through partially-worked examples. For example, during the initial lesson entitled 'Data Science and the Nature of Data', students read data files into table-like structures called data-frames, selected columns from dataframes, and filtered dataframe rows using a value; and when learning about Plotting, students practiced making scatter plots. Figures 2 & 3 show part of a lesson on scatter plots in Jupyter Notebook. This excerpt is from a morning notebook that was presented the first week of the internship and contained detailed instructions for using the interface and scaffolding methods used to support the learning of new information.

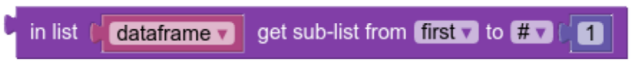
Embedding the learning materials and associated activities within Jupyter Notebook was helpful because it displayed related content together, supporting the spatial contiguity effect, which argues that people learn better when pictures and words are shown spatially near one another, rather than far apart (Johnson & Mayer, 2012; Schroeder & Cenkci, 2018). Since Jupyter Notebook provided a single source for both the workspace and necessary study materials, students did not need to manage different technologies or texts. If students had problems, they could scroll up, find the necessary information, and reattempt their code. As opposed to a static textbook, this integrated approach affords a more iterative problem-solving experience that is important when novices are first learning to code in data science.

Jupyter Notebook Afternoon Assignments

Following the morning exercise, students took a break for lunch, and a faculty member joined them once a week for lunch via Zoom. In the afternoon, instructors tasked students with completing an isomorphic problem; that is, a problem similar in form to the morning worked example. Students could consult their morning notebook, as well as an expert notebook as they worked. At this point, students were tasked with solving problems on their own as a way to facilitate learning transfer. The isomorphic notebooks also contained reflection questions asking students to make predictions and address hypotheticals. Since they were all assigned the same problem, they were encouraged to consult with their

Dataframes as a list of rows

There are many things we can do with dataframes. One thing we can do is get specific rows, which are our datapoints. Using the LIST menu in the Blockly palette, click on the `in list listVariable get sub-list from to`. Next change `listVariable` to `dataframe`, the first `#` to `first`, and drop a number block `123` from MATH in the second `#`, then change the value of the number block to `1`. Your blocks should look like this:



In the future, we'll abbreviate this as:

- `in list dataframe get sub-list from first to 1`

Then ▶ or Shift + Enter

:

:

	Height	Age	Weight
0	161	50	53

As you can see, the output is only the first row of the dataframe.

Try it again (i.e. copy the blocks, select the cell below, and paste the blocks in the Blockly workspace), but this time, change the `1` to a `2`:

- `in list dataframe get sub-list from first to 2`

Then ▶ or Shift + Enter

[7]:

[7]:

	Height	Age	Weight
0	161	50	53
1	161	17	53

Now the output is the first two rows of the dataframe. We could get arbitrary rows of the dataframe by starting at a different number and ending at a different number. Sometimes people call this a **slice**.

You might be wondering at this point what the numbers are on the left side of our output. They aren't data in our dataframe - they are actually row identifiers `pandas` has automatically assigned to our datapoints. In pandas, these are called an `index`.

When the index is numeric, it's easy to see if you got the rows you wanted. Just remember that the index starts at 0 by default rather than 1 (computers count from zero!).

FIGURE 4–5. Jupyter Notebook Lesson with Embedded Problem—Parts 1 and 2.

peers and their instructors if they had questions or ran into any issues or roadblocks when trying to solve the afternoon problems. Typically, students would stay on the standing Zoom meeting to chat and ask each other questions as they worked on the afternoon notebook. There was always at least one instructor, but typically more, in the online instant-messaging tools (e,g. Slack) to help with any questions or technical difficulties. Figures 4 & 5 show an example of a simple linear regression lesson with an embedded problem in the Jupyter format.

Peer Reviews

Following the isomorphic notebook session, students would work in small teams (2-3) to review their fellow students' notebooks for that afternoon. The design team reasoned that working together in this way allowed students to gain learning benefits of both peer reviews and team-based work. To guide their review, students were instructed to compare solutions to their own and reflect on the advantages/disadvantages of both. These instructions were designed to promote justification as a data science practice, since many data science problems are ill defined. Students were similarly asked to review the answers to reflection questions embedded in the notebooks.

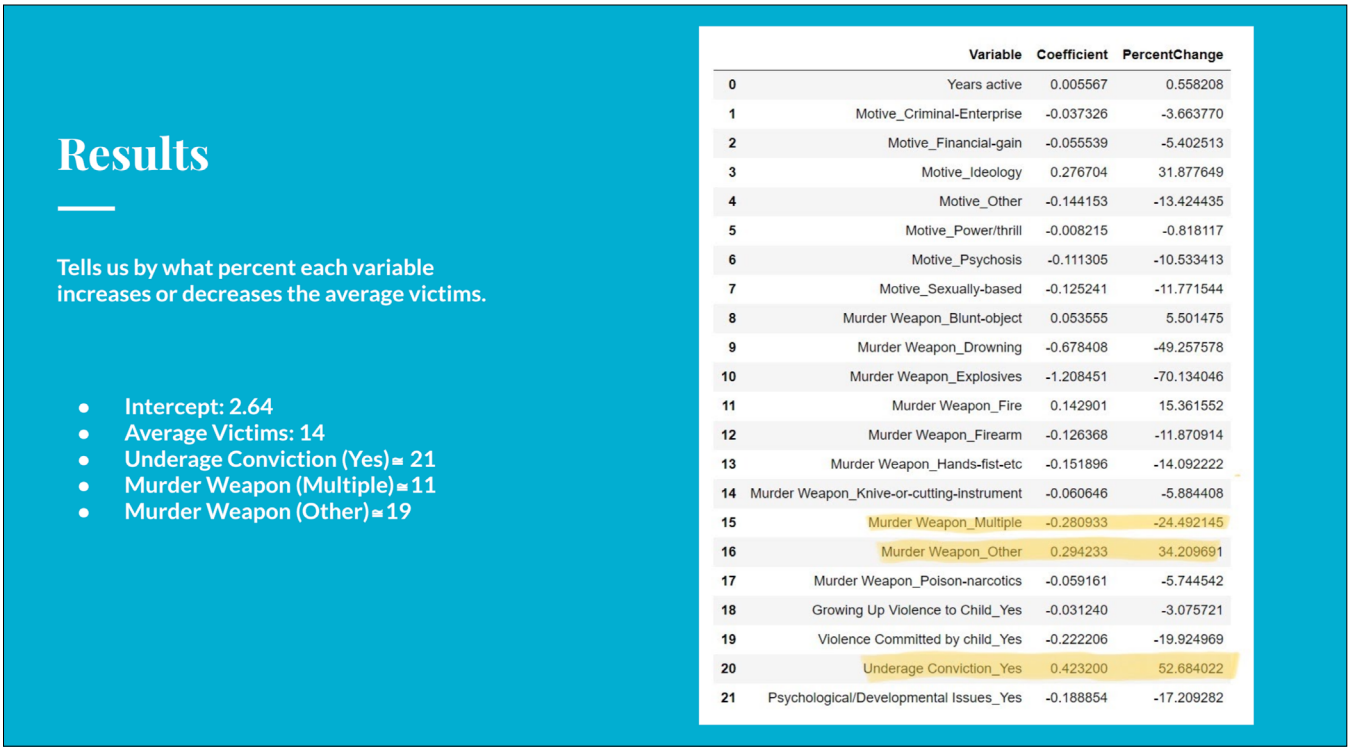
Reflection Hour

In the last hour of each day of the internship, students attended a reflection session conducted by one or more faculty members. Research has shown that reflection is important in facilitating meaningful learning among students, especially in complex problem-solving (Veine et al., 2020; Billing, 2007; Costa & Kallick, 2008; Lin, et al., 1999; Moon, 2004); therefore, we wanted to make this a strategic part of the learning experience. This was especially important given our shift to online, which required that learners engage in more independent problem-solving. Although these tended to vary by who was leading the reflection, discussion topics included what was learned, how new concepts related to prior topics, what strategies were employed during problem-solving, and comparing solutions and answers to reflection questions. The goal of the reflection was to finalize schema building and clarify any outstanding issues they had. Since learning from reflection is most effective when students compare and contrast their learning to those of others (Dewey, 1933; Rud, et al., 2009; Simpson, et al., 2005), our reflection hours were designed to be group-based, allowing students to not only learn by reflecting on their own experiences during the problem-solving process, but by listening to the reflections of their peers as well.

SUMMER INTERNSHIP INSTRUCTIONAL ACTIVITIES—PART 2

For the last three weeks of the summer internship, we divided the students into two teams, with each team working to solve a different data science problem using what they had learned during the first five-week phase of instruction. The original plan for July was to gather data from local community organizations and help them find solutions to address a current problem or need. However, due to issues with organization loss of data and last-minute withdrawals due to COVID-19, we transitioned toward more student-selected projects based on their interests. Students were given a day to brainstorm ideas, research topics, gather relevant data, and present their ideas to instructors. The two topics that students came up with, based on their own interests and that instructors approved, were movie recommendations and predicting victims of killers. This was a way of allowing students to get hands-on experience finding data on their own and using the data science concepts, tools, and techniques they learned in class to answer a variety of real-world questions.

This second phase had a more on-demand structure when compared with the first phase. However, just like in the first part of the summer, instructors were available via Slack and email throughout the day to answer questions and support students as needed. Specifically, each team was assigned a mentor and co-mentor(s), based on their knowledge and



	Variable	Coefficient	PercentChange
0	Years active	0.005567	0.558208
1	Motive_Criminal-Enterprise	-0.037326	-3.663770
2	Motive_Financial-gain	-0.055539	-5.402513
3	Motive_Ideology	0.276704	31.877649
4	Motive_Other	-0.144153	-13.424435
5	Motive_Power/thrill	-0.008215	-0.818117
6	Motive_Psychosis	-0.111305	-10.533413
7	Motive_Sexually-based	-0.125241	-11.771544
8	Murder Weapon_Blunt-object	0.053555	5.501475
9	Murder Weapon_Drowning	-0.678408	-49.257578
10	Murder Weapon_Explosives	-1.208451	-70.134046
11	Murder Weapon_Fire	0.142901	15.361552
12	Murder Weapon_Firearm	-0.126368	-11.870914
13	Murder Weapon_Hands-fist-etc	-0.151896	-14.092222
14	Murder Weapon_Knive-or-cutting-instrument	-0.060646	-5.884408
15	Murder Weapon_Multiple	-0.280933	-24.492145
16	Murder Weapon_Other	0.294233	34.209691
17	Murder Weapon_Poison-narcotics	-0.059161	-5.744542
18	Growing Up Violence to Child_Yes	-0.031240	-3.075721
19	Violence Committed by child_Yes	-0.222206	-19.924969
20	Underage Conviction_Yes	0.423200	52.684022
21	Psychological/Developmental Issues_Yes	-0.188854	-17.209282

FIGURE 6. Results Slide from Team Project on Predicting Victims of Serial Killer.

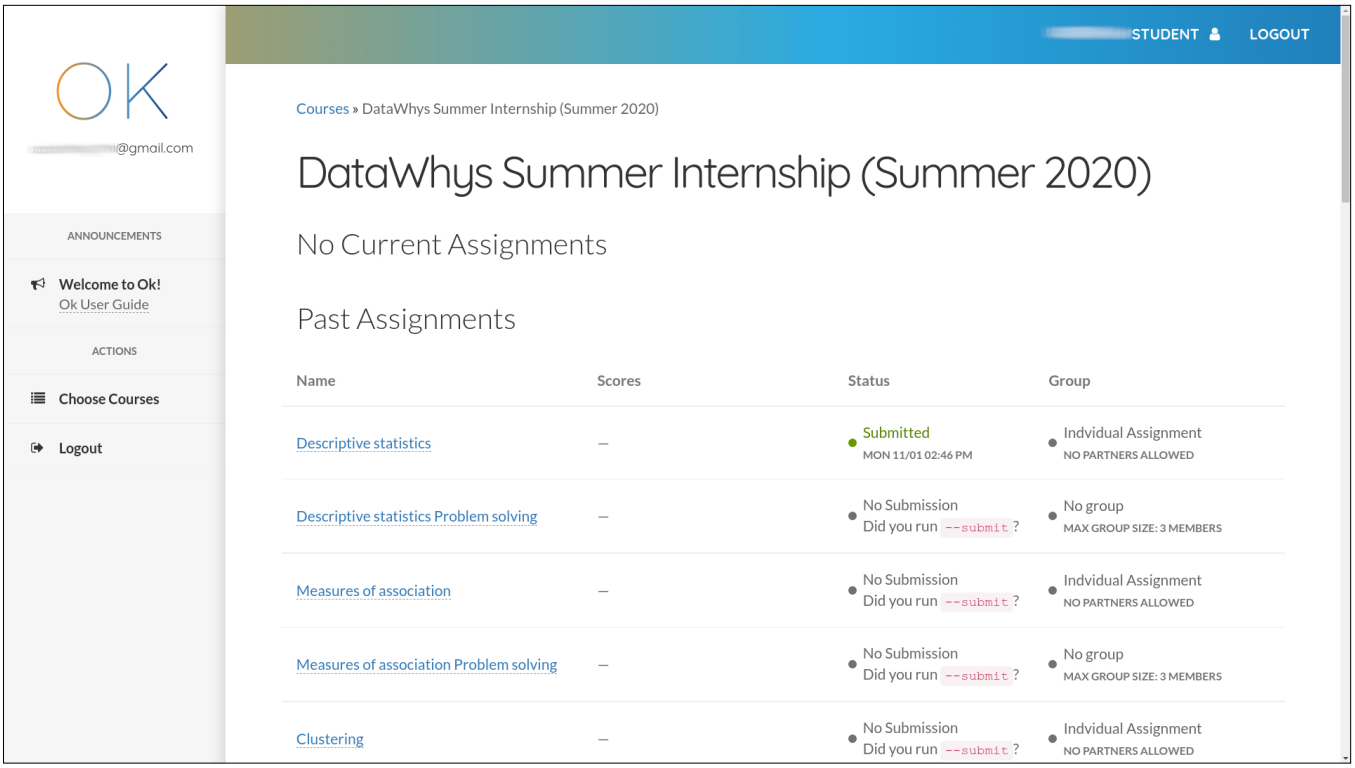


FIGURE 7. OKpy—Student View.

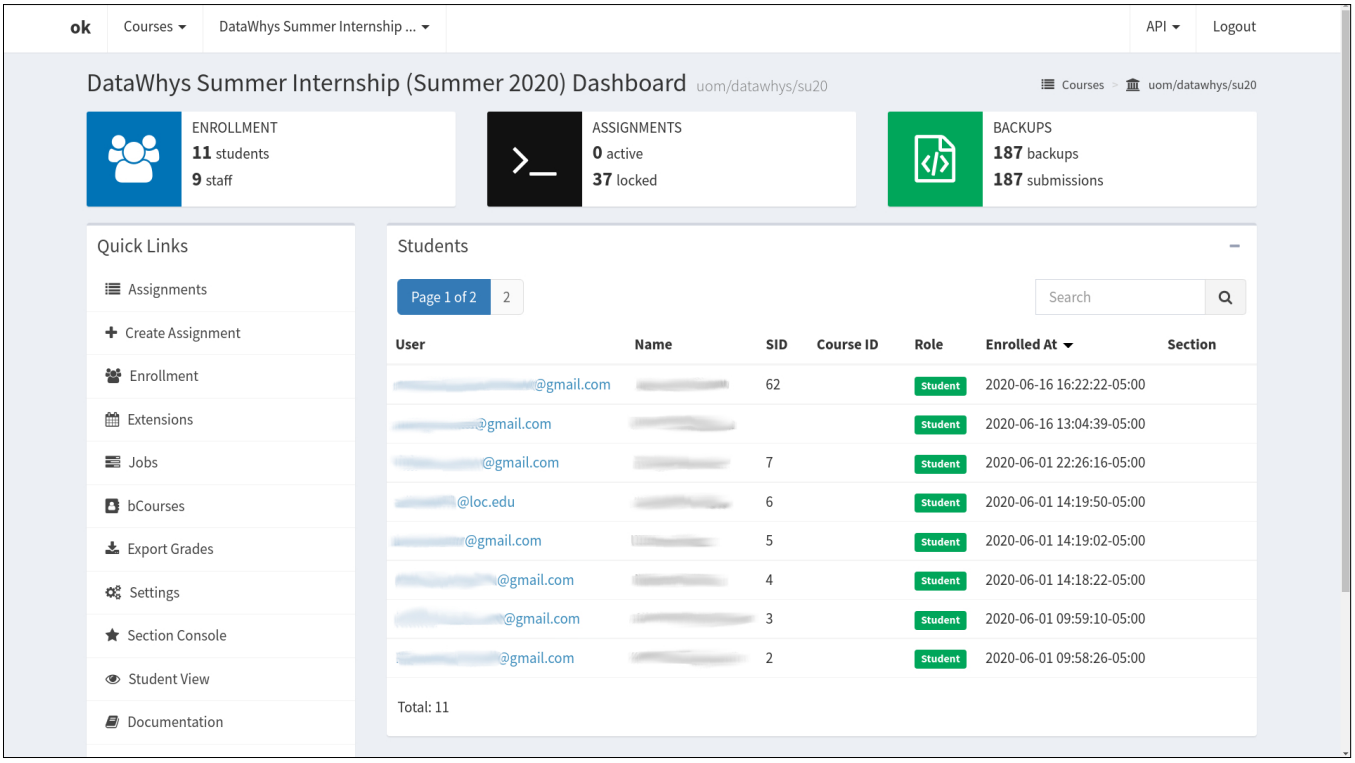


FIGURE 8. OKpy—Instructor View.

interest surrounding the particular topic. Instructors also continued to host faculty lunches and check in on Zoom two to four times a day, depending on how things were progressing. At the end of the second summer session, teams of students were asked to give a live video presentation of their project results to instructors and graduate students. Students on one team found that despite their predictions of the importance of childhood trauma and motivation to kill, the only significant predictors of victim count were having an underage conviction, use of multiple murder weapons, and use of exotic murder weapons." Figure 6 is a sample slide from one team's presentation that shows the results from the team that worked on predicting victims of serial killers.

DESIGN TENSIONS AND DECISIONS

In the following section, we detail the unique challenges of teaching data science using an online learning format. Specifically, they include: Challenges of Data Science Instruction Using Existing Computer Science Tools, Instructional Design of Materials that Supported More Self-Directed Learning, and Collaborative Problem-Solving.

Challenges of Data Science Instruction Using Existing Computer Science Tools

Due to the dramatic shift to online instruction, we needed to leverage a learning management system for various aspects of the course. However, the design team needed to select one that could uniquely handle the complexities of data science learning objectives: manipulating, storing, and running the data science code. The design of the original face-to-face experience sought to use GitHub Classroom solely as a place where instructors could assign notebooks to students and students could turn in their work. However, the online approach caused the team to reconsider supports for other important elements of instruction, such as daily tasks, centralized communication portals, and resource sharing. The team further researched if GitHub Classroom could accommodate more instructional needs. Unfortunately, GitHub Classroom did not have the learning management system (LMS) capabilities needed for online classes. Also, we found it to be focused heavily on teaching GitHub in the context of general programming, which was not a goal of this project.

In lieu of no comprehensive learning technology to fully support data science in a remote context, the team determined that the best design approach would be a hybrid solution using multiple technologies. Students used a Google Calendar schedule on the internship portal to know what they were supposed to be working on at any given time. The schedule had links to Zoom for standing meetings and reflection sessions. For Jupyter Notebooks, the schedule contained special links for distributing notebooks using the Jupyter Hub. Once clicked, these links would take students to the Jupyter Hub server page, log them in, retrieve the relevant Jupyter notebook from GitHub, and open the notebook. Students would then progress through the notebook in the normal way, typically using Blockly. Each notebook

FIGURE 9. Web Portal Page.

contained an instruction to submit the notebook to OKpy for instructor review and feedback (see Figures 7 & 8). OKpy is a simple web-based submission system developed by the University of California Berkeley for their large programming and data science courses with capabilities similar to a drop box; however, a key feature for our purposes is that OKpy will natively render notebooks, including plots, making it easy for instructors to review submissions without opening them in Jupyter.

Given that there was no central solution, the design team decided to develop other tools used to support the unique nature of online learning for data science. For example, the Web portal page discussed earlier (see Figure 9) provided a daily schedule and links to notebook lessons, meetings, and other useful content. University email was used for official communications, whereas day-to-day communication utilized Slack with separate channels for announcements, chat, help, and team projects. Finally, Zoom facilitated synchronous classroom tasks, including lectures, demonstrations, discussions, screen sharing, and other collaborative teamwork. Again, this multifaceted approach was needed given the lack of a tool that supported requisite communication and interaction with the data science concepts within the course.

Instructional Design of Materials that Supported More Self-Directed Learning

Although the original intent of the Datawhys internship was to more directly scaffold learners in a face-to-face lab experience, the shift toward online learning due to COVID-19 necessitated the rethink and redesign instructional materials through the lens of self-directed learning. Because learners were focused on various aspects of data science (the statistical concepts, creation of code, interpretation), the design team created instructional materials using the Jupyter platform for each specific topic. As stated earlier, this tool allows individuals to run various programming languages. As such, the design team constructed individual notebooks for various data science topics and embedded content within the notebooks.

This design decision was unique for multiple reasons. Rather than assigning learning content across various sources, the design team was able to design lessons with materials related specifically to the topics of this course in one central location and without any unnecessary information. Second, using Jupyter Notebook as a design platform helped us mitigate cognitive load and reduce textbook costs because the learning materials could be accessed in a single, web-based, and open-source format. Finally, developing the learner materials using Jupyter Notebook allowed us to embed various forms of media, especially those that were responsive to the embedded code. That is, students could view worked examples in the form of media, while also dynamically interacting with the data science visualization. Figure 10 shows an excerpt from one of our Jupyter Notebook lessons.

By expanding Jupyter Notebook as a platform, as opposed to just a data science processing application, the design decision elucidated unique design challenges. First, the rapid decision to move online required us to design the lesson materials for the notebooks very quickly and with little testing. The creation of the notebook material was complex and involved two to three instructors working on each notebook to cover didactic content, Python code, and conversion to blocks. In some cases, the Python code and conversion to blocks were done together, but not always. Although the singular approach helped overcome the lack of an LMS, the design team had to weave various forms of media into a single source. We struggled with a "wall of text" scenario that involved learners having to scroll through copious amounts of text, which could easily lead to cognitive overload. Therefore, we were challenged to make sure the lessons within the notebooks contained the right balance of information and interaction suitable for a wide range of learners, while at the same time, managing the learners' working memory.

To maintain coherence of the overall program, the lead principle investigator developed a list of 20 data science topics for the five weeks of the initial boot camp phase. Once the final topics were discussed/agreed upon, instructors signed up to create notebooks in the didactic, programming, or blocks roles. Although the team had discussed a general design for the notebooks based on worked examples and interweaving instruction with problem solving, instructors began working on their respective notebooks independently without strong guidance or a working example. Despite the lack of guidance, the resulting notebooks were remarkably similar in style, differing mostly in terms of the treatment given particular aspects of the content. For example, instructors with computer science backgrounds tended to emphasize algorithmic and procedural aspects of the content, whereas instructors with statistics backgrounds tended to include mathematical formulas and refer to them in text. In order to maintain a consistent treatment of the material, the lead principle investigator acted as the final editor but nevertheless did not override instructor decisions to include more nuanced material. As instructors continued to produce and review each other's work, a more consistent treatment of the material began to emerge.

Collaborative Problem-Solving

Because the program was originally designed to be a face-to-face experience that emphasized iterative peer (faculty, student) support, the online approach caused us to rethink what design strategies and limited tools best supported collaborative learning of data science at a distance. As noted earlier, we had considered domain-specific tools for learning

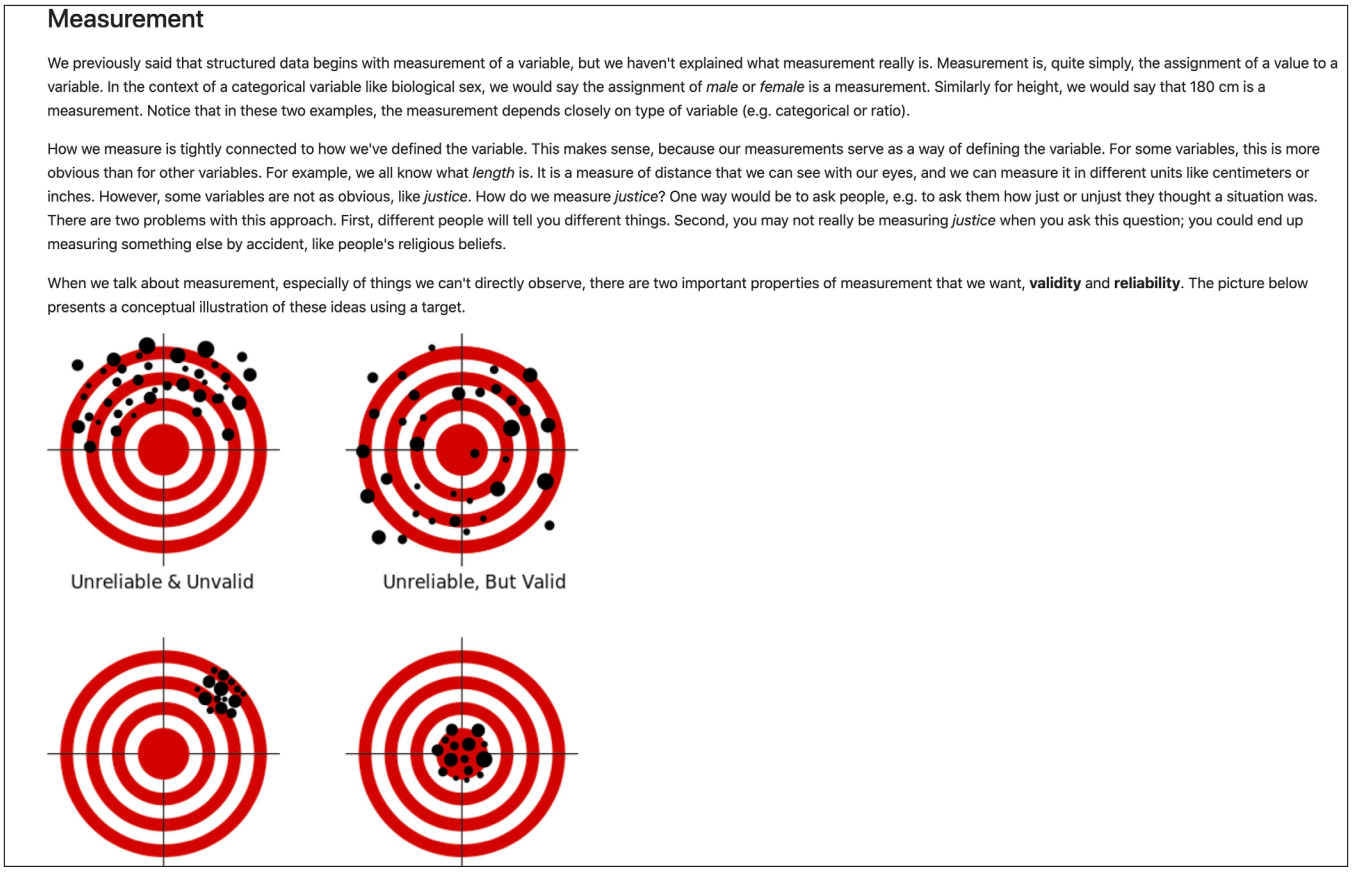


FIGURE 10. Jupyter Notebook Lesson Excerpt with Graphic.

(e.g., Github Classroom, Jupyter), but none supported the collaborative learning aspect that was important to teaching data science for learners with varying levels of background knowledge. Therefore, the design team decided to employ Slack, an instant messaging tool often used in practice.

The use of Slack brought both benefits and challenges to the project. The instant messaging of Slack (see Figure 11) allowed components of both synchronous and asynchronous learning. The approach also allowed other elements that were important for the problem-solving piece, such as file sharing. However, the limitations of Slack prevented implementation of collaborative pedagogies like pair programming that typically require joint control of physical artifacts. In pair programming, one student acts as the "driver" and the other plays the role of "navigator," which allows the driver to focus on the task of typing code and the navigator to focus on catching mistakes and keeping the driver on track. The key practice of switching roles, which is as simple in a physical collocated environment as switching seats, becomes much more technically challenging when students are remote. As with the other challenges, the impact of COVID19 highlights the limited pedagogical tools needed to teach data science effectively, especially in an online format.

DESIGN FAILURES AND REFLECTIONS

Three main themes emerged that will impact the future iterations of our design-based research approach, based on observations of the students and review of their work. First of all, some students were not finishing the notebooks. Each day (Monday—Thursday), instructors provided the students with an oral overview, written instructions, and a worked example for that day's topic. In the afternoon, instructors gave students a second assignment to complete on their own using the same principles as the worked example from earlier in the day. We expected an initial learning curve; however, as time progressed, the design team observed that many students still were not finishing, and the concern was that the students were not mastering one lesson before moving on to the next. Even though the lessons were scaffolded, we could have reconsidered our fading strategy to better support the varying levels of prior knowledge as they created schemas for long-term memory. One instructor expressed a concern that "the topics were coming on too fast" and that there was an "abrupt leap to the practice problem."

Secondly, students were not as engaged in the group discussions as we would have liked. Instructors additionally observed a lack of participation in daily reflection meetings. As we consider this design challenge, this may be due to the

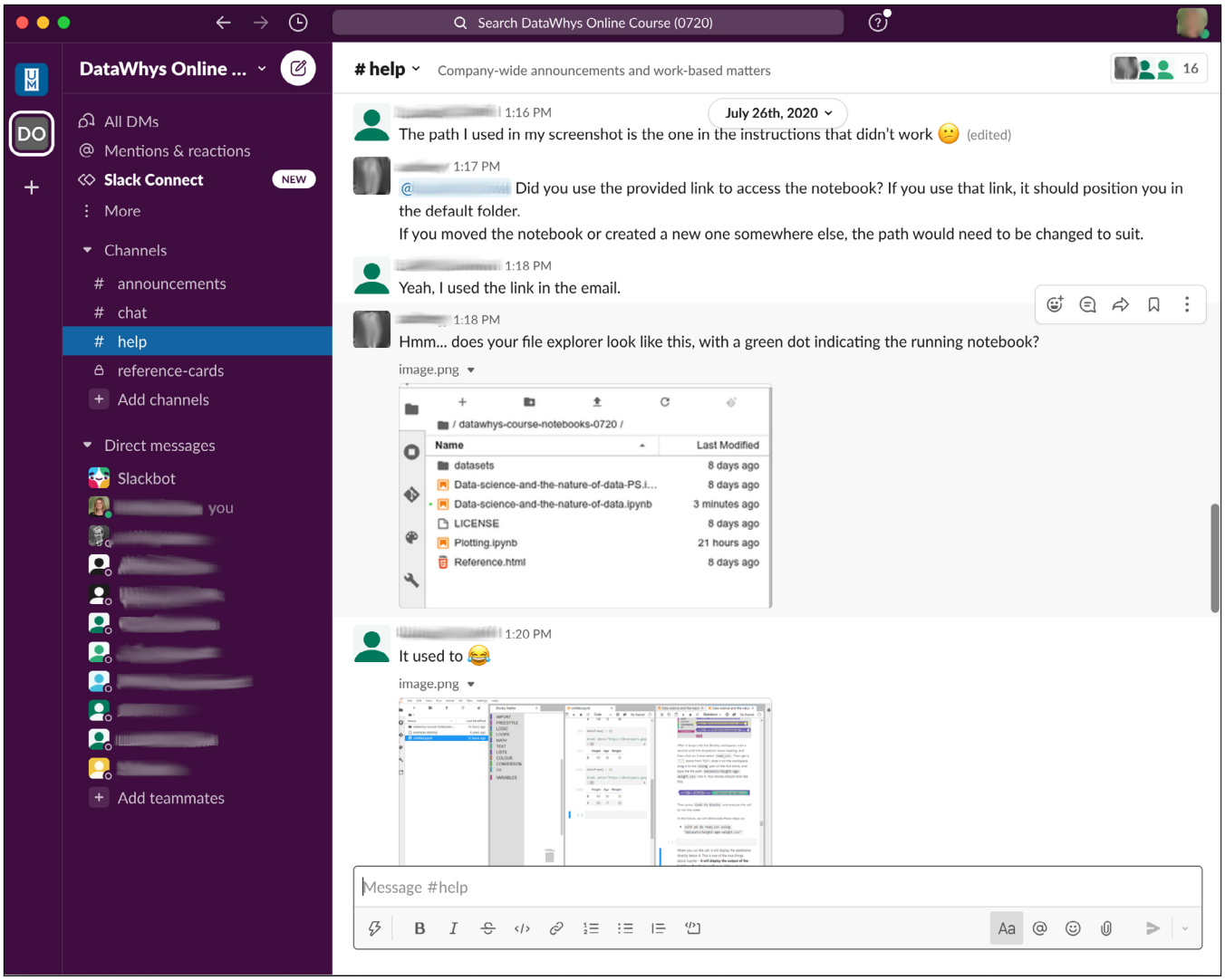


FIGURE 11. Collaborative Problem-Solving in Slack.

challenge of learning data science, which includes both the statistical concepts (e.g., lasso regression) and the computational thinking components (e.g., algorithmic thinking, debugging, etc). Although students seemed to understand the basic concepts, they appeared to struggle to execute the more technical aspects required for the programming lessons. As the course progressed, even students who were typically the most engaged had trouble answering some questions about the lesson, whether the questions were based on theory or application.

Third, students were having difficulty with the technologies used, particularly Jupyter Notebook and Blockly (Figure 6 shows the Jupyter Notebook with Blockly interface used). Technical issues interfered with students' efforts to work through the notebooks. To add to this, some students were working on smaller screens (e.g., 1366x768), which inhibited them from easily navigating the interface. This could pose a considerable design challenge when developing online instruction for teaching data science, as the tools needed to teach data science might not be easily scalable to smaller screens. Additionally, small screens can create extraneous cognitive load when students are attempting to use worked examples to solve new problems, but cannot fit both the worked example and new problem on the screen at the same time.

FUTURE DESIGN STRATEGIES

Instructors were able to incorporate some improvements "on the fly," such as making sure that instructors were providing a consistent experience for the students. For example, instructors agreed to provide an overview of each day's lesson at the start of every morning. They also discussed new strategies for facilitating the afternoon reflection sessions with students. A related change implemented during the summer internship was to provide the correct answers to the morning problems before the afternoon sessions. This helped students to know if they were on the right track going into the afternoon problem. It also provided students

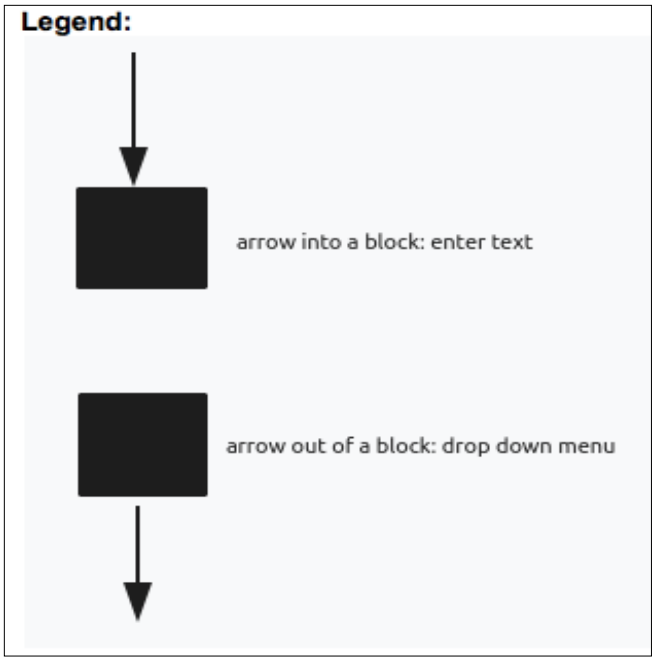


FIGURE 12. Legend Explaining Direction of Arrows on Reference Cards.

with reassurance that they were taking the right approach to the problem before starting a similar problem in the afternoon, or it helped them to see what they got wrong in the morning so that they would not make the same mistakes in the afternoon. For students who did not finish the morning notebook, the correct answers were even more important as a reference for isomorphic problems in the afternoon.

As we move forward with our design-based research project, the experience provides insight into the challenges of learning data science remotely; that is, challenges with communication among students and instructors, weaknesses of technologies available, or equal access to the right educational tools needed to learn data science online. Specific improvements planned include the following: 'ramp up' activities; scope of the program; scaffolding strategies, instructional design of the learning materials (Jupyter Notebooks); providing earlier feedback regarding the correct answers to worked examples; formal usability testing of the primary user interface; the possible addition of pair programming; and the addition of reference cards and other aids to supplement learning and reduce cognitive load.

One of the themes mentioned earlier was the lack of a comprehensive tool that supported both the technical and pedagogical aspects of data science. As such, we hope to further refine the Jupyter notebooks from an instructional design and UX perspective. For example, we plan to perform usability testing on the tools used to create the notebooks, and specifically on using Jupyter Notebook with the Blockly plug-in, as this software seems to have given novice students the most trouble. The initial usability test will involve

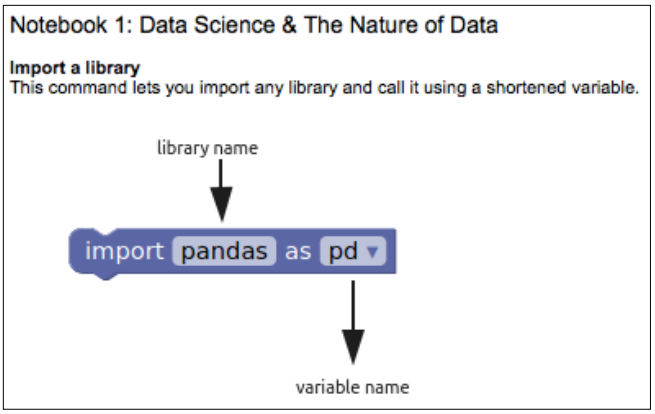


FIGURE 13. First Reference Card for Notebook 1 Showing Command to Import a Library.

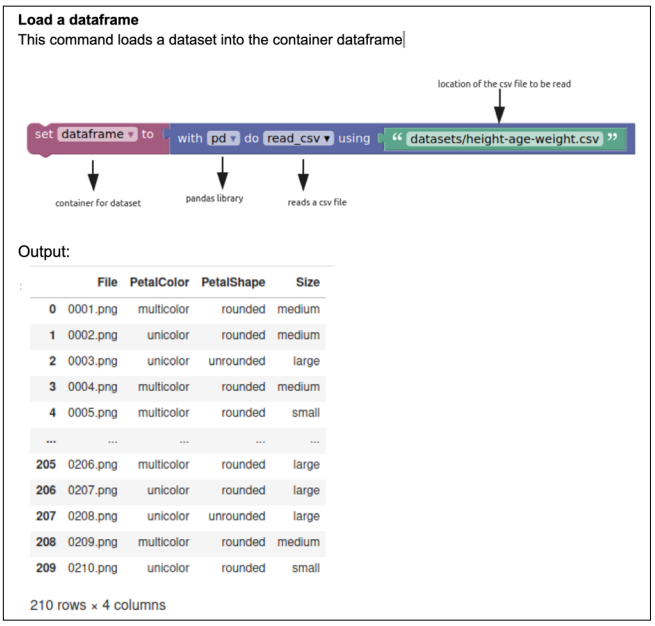
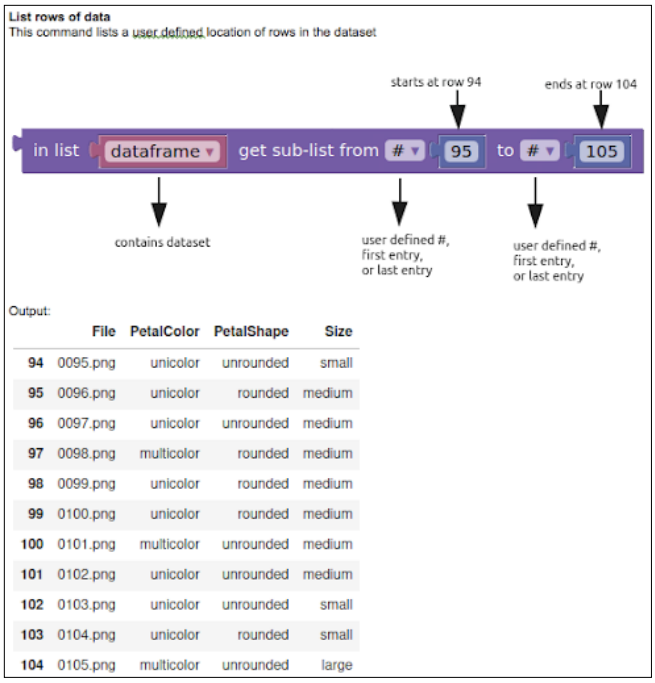


FIGURE 14. Reference Card Showing Command to Load a Dataframe.

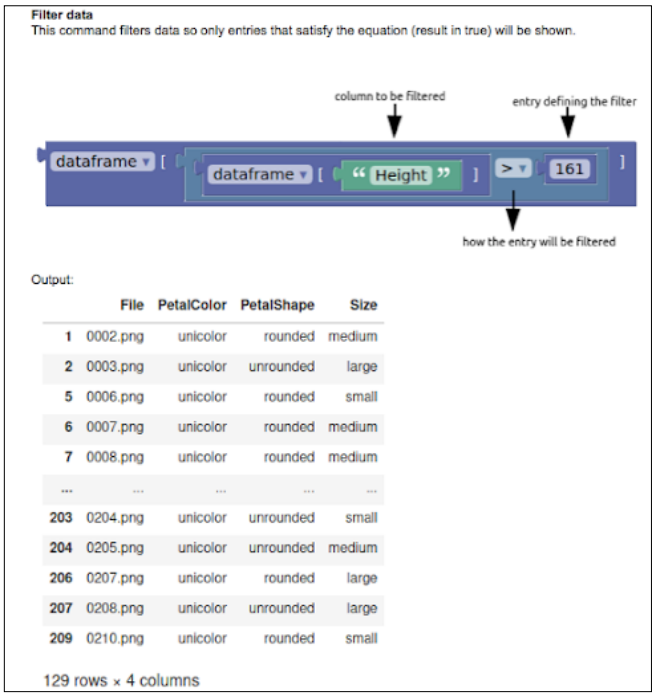
five participants who will be asked to perform steps required to complete four entry-level coding tasks: 1) Create a block and convert it to code; 2) Print a given word five times; 3) Edit code in the Jupyter Notebook without using blocks; and 4) Explore two different ways to copy and paste into different cells. Again, we will use the results of this study to improve the interface design of the program, as well as to inform the instructional design to the extent to which it is impacted by the user interface.

As stated earlier, pair programming was another idea that emerged during this process. When programmers have a difference in skill, pair programming can embody the two main principles of Vygotsky's (1978) socio-cultural theories of learning: the more knowledgeable other (MKO) and the zone of proximal development (ZPD). As programmers collaborate to solve problems together, the more novice programmer



	File	PetalColor	PetalShape	Size
94	0095.png	unicolor	unrounded	small
95	0096.png	unicolor	rounded	medium
96	0097.png	unicolor	unrounded	medium
97	0098.png	multicolor	rounded	medium
98	0099.png	unicolor	rounded	medium
99	0100.png	unicolor	rounded	medium
100	0101.png	multicolor	unrounded	medium
101	0102.png	unicolor	unrounded	medium
102	0103.png	unicolor	unrounded	small
103	0104.png	unicolor	rounded	small
104	0105.png	multicolor	unrounded	large

FIGURE 15. Reference Card Showing Command to List Rows of Data.



	File	PetalColor	PetalShape	Size
1	0002.png	unicolor	rounded	medium
2	0003.png	unicolor	unrounded	large
5	0006.png	unicolor	rounded	small
6	0007.png	unicolor	rounded	medium
7	0008.png	unicolor	rounded	medium
...	...	...	...	...
203	0204.png	unicolor	unrounded	small
204	0205.png	unicolor	unrounded	medium
206	0207.png	unicolor	rounded	large
207	0208.png	unicolor	unrounded	large
209	0210.png	unicolor	rounded	small

129 rows × 4 columns

FIGURE 16. Reference Card Showing Command to Filter Data.

learns from the more experienced peer programmer. Furthermore, pair programming is conducive to learning via the ZPD theory (Vygotsky, et al., 1978) by working with help from a more experienced peer and through the use of technology and tools provided to both programmers. Pair programming also serves to reduce cognitive load by allowing one programmer to focus on monitoring for errors, while the other can focus on writing and iterating code. Research has shown further benefits of pair programming to include understandability and maintainability of code and design (Alves & Berente, 2016; Plonka et al., 2015; Vanhanen & Korpi, 2007); higher quality and lower defect rates (Jensen, 2003; Plonka et al., 2015; Phongpaibul & Boehm, 2006); and increased knowledge transfer (Katriou & Tolias, 2009; Plonka et al., 2015; Sanders, 2002; VanDeGrift, 2004; Vanhanen & Korpi, 2007; Vanhanen & Lassenius, 2005). If we are able to offer future courses in person, we anticipate the limited online tools that facilitate pair programming will be less of an issue. However, if the course continues to operate solely online, it is likely this will add additional complexity. We will need to work toward a solution, possibly incorporating Zoom functionality, which allows one student to see what the other is typing and that lets students switch control over the same document.

As stated earlier, one of the problems we discovered early on was that the students came into the class with different levels of knowledge related to computer science and data science. For those students with very little to no experience with coding, it became apparent that they needed more time to learn the basic coding commands before being able to move through the lessons within the allotted time. Early design plans for the larger Datawhys project included the use of intelligent assistants to aid teachers and guide students through the lessons, and this is something that we plan to develop for a future phase of the project. The idea is that the intelligent pedagogical assistant will help learners track and understand the alternatives they encounter during open-ended data science problem-solving. The intelligent assistant will extend a continuum of adaptive support provided by worked examples, intelligent tutoring systems, and open problem-solving environments by helping learners understand what options exist for particular goals and why some options might be preferred.

One idea for addressing this issue in future project phases, at least in part, was to create reference cards outlining the various commands, the purpose of each command, the blocks and associated variables used to perform each command, and the resulting outputs of each command. In future phases of the project, we will provide these reference cards, which will include text aligned appropriately with associated graphical blocks for spatial contiguity and to reduce the need for visual scanning (Mayer & Moreno, 2010), for students to use as "cheat sheets" whenever they need them throughout the course. Ideally, this will reduce cognitive load, as students will have these cards to easily reference as needed and not have to spend considerable mental effort searching through previous worked examples to figure out what commands to use to perform certain tasks. This will afford students more cognitive processing space for critical thinking and problem-solving related to the given data science problem. Figures 12-16 show the reference cards

created by one of the graduate assistants on the team for one of the lessons written by instructors. In addition to the reference cards, students will be further supported with updated lessons/worked examples that include instructions regarding when to use certain commands.

CONCLUSION

This design case highlights the unique intersection of the STEM domain and informal learning using an internship strategy. First, it underscores how many existing tools lack the design features to facilitate a fully online internship in computer science and data science, specifically, the technical components needed for learning in this domain Second, it identifies how COVID-19 affected education beyond the traditional K-12 or higher education context; in this case, informal learning for HBCU students. This speaks to the far-reaching disruption of COVID-19 and its impact on programs designed for equitable learning experiences.

Furthermore, this design case is an example of the iterative nature of instructional design that is crucial to creating effective learning programs, and it emphasizes the need for flexibility in terms of instructional design methods and technologies. As outlined earlier, we have already begun to take the lessons our team has learned from the summer intern program experience to improve upon the initial curriculum design. As we design and develop future iterations, we will learn from those, and then ultimately design a successful artificial intelligence-assisted pedagogical system that supports computational thinking at a distance for students who desire to learn data science, regardless of their background or socioeconomic situation.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1918751. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Alves De Lima Salge, C., & Berente, N. (2016). *Pair programming vs. solo programming: What do we know after 15 years of research?* 49th Hawaii International Conference on System Sciences, 5398–5406. https://doi.org/10.1109/HICSS.2016.667

Billing, D. (2007). Teaching for transfer of core/key skills in higher education: Cognitive skills. *Higher Education, 53(4)*, 483-516. https://doi.org/10.1007/s10734-005-5628-5

Costa, A.L., & Kallick, B. (2008), Learning through reflection. In A.L. Costa & B. Kallick (Eds.), *Learning and leading with habits of mind* (ch. 12). Association for Supervision and Curriculum Development.

CSLI Summer Internship Program | Center for the Study of Language and Information. (n.d.). https://www-csli.stanford.edu/csli-summer-internship-program

Data Science for Social Good Summer Fellowship—Carnegie Mellon University. (n.d.). https://www.dssgfellowship.org/

Jensen, R. (2003). A pair programming experience. *CrossTalk: A Journal of Defense Software Engineering, 16*(3), 22–24.

Johnson, C., Mayer, R.E. (2012). An eye movement analysis of the spatial contiguity effect in multimedia learning. *Journal of Experimental Psychology, 18*(2), 178-191. https://doi.org/10.1037/a0026923

Katriou, S. & Tolias, E. (2009). *From twin training to pair programming*. Proceedings of the 2nd India Software Engineering Conference, 101–104. https://doi.org/10.1145/1506216.1506235

Ladson-Billings, G. (2006). From the achievement gap to the education debt: Understanding achievement in U.S. schools. *Educational Researcher*, *35*(7), 3–12. https://doi.org/10.3102/0013189X035007003

Lin, X., Hmelo, C., Kinzer, C. K., et al. (1999). Designing technology to support reflection. *Education Training Research and Development, 47*(3), 43-62. https://doi.org/10.1007/BF02299633

Mayer, R. E., Moreno, R. (2003). Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist, 38*(1), 43-52. https://doi.org/10.1207/S15326985EP3801_6

Moon, J. (2004). A handbook of Reflective and Experiential learning: Theory and practice. Routledge. https://doi.org/10.4324/9780203416150

Morton, T. R. (2020). A phenomenological and ecological perspective on the influence of undergraduate research experiences on Black women's persistence in STEM at an HBCU. *Journal of Diversity in Higher Education*. Advance online publication. https://doi.org/10.1037/dhe0000183

OK, (n.d.). https://www.okpy.org

Palmer, R. T., Maramba, D. C., & Dancy, T. E. (2011). A qualitative investigation of factors promoting the retention and persistence of students of color in STEM. The Journal of Negro Education, 80(4), 491–504. http://www.jstor.org/stable/41341155.

Morreale, P., Burnett, M., Gates, A., Cossa, J., & Amato, N. (2011). *REU-in-a-box: Expanding the pool of computing researchers.* National Center for Women & Information Technology. http://www.ncwit.org/reubox

Phongpaibul, M. & Boehm, B. (2006). *An empirical comparison between pair development and software inspection in Thailand.* Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, 85–94. https://doi.org/10.1145/1159733.1159749

Plonka, L, Sharp, H., van der Linden, J., & Dittrich, Y. (2015). Knowledge transfer in pair programming: An in-depth analysis. *International Journal of Human-Computer Studies, 73*(1) pp. 66–78. https://doi.org/10.1016/j.ijhcs.2014.09.001

Rud, A. G., Garrison, J., Stone, L. (2009). John Dewey at 150: Reflections for a new century. *Education and Culture: The Journal of the John Dewey Society, 25*(2).

Sanders, D. (2002). Student perceptions of the suitability of extreme and pair programming. In *Extreme Programming Perspectives*. (ch. 23). Addison-Wesley.

Schroeder, N. L., Cenkci, A. T. (2018). Spatial contiguity and spatial split-attention effects in multimedia learning environments: a meta-analysis. *Educational Psychology Review, 30*(3), 679-701. https://doi.org/10.1007/s10648-018-9435-9

Simpson, A., & Maltese, A. (2017). "Failure is a major component of Learning Anything:"The role of failure in the development of STEM professionals. *Journal of Science Education and Technology, 26*(2), 223–237. https://doi.org/10.1007/s10956-016-9674-9

Simpson, D. J., Jackson, M.J.B., & Aycock, J.C. (2005). *John Dewey and the art of teaching.* Sage Publications. http://dx.doi.org/10.4135/9781452232386

Stehle, S. M., & Peters-Burton, E. E. (2019). Developing student 21st century skills in selected exemplary inclusive STEM high schools. *International Journal of STEM Education*, *6*(1). https://doi.org/10.1186/s40594-019-0192-1

VanDeGrift, T. (2004). *Coupling pair programming and writing: learning about students' perceptions and Processes* Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, 2–6. https://doi.org/10.1145/971300.971306

Vanhanen, J. & Korpi, H. (2007). *Experiences of using pair programming in an agile project* [Paper presentation]. 40th Annual Hawaii International Conference on System Sciences, 274b–274b. https://doi.org/10.1109/HICSS.2007.218

Vanhanen, J. & Lassenius, C. (2005). *Effects of pair programming at the development team level: An experiment*. 2005 International Symposium on Empirical Software Engineering, 336-345. https://doi.org/10.1109/ISESE.2005.1541842

Veine, S., Anderson, M. K., Andersen, N. H., Espenes, T.C., Søyland, T.B., Wallin, P. & Reams, J. (2020). Reflection as a core student learning activity in higher education—Insights from nearly two decades of academic development. *International Journal for Academic Development*, *25*(2), 147-161. https://doi.org/10.1080/1360144X.2019.1659797

Vygotsky, L.S. (1978). Mind in society: The development of higher psychological processes. Harvard University Press. https://doi.org/10.2307/j.ctvjf9vz4