

Value of Software Carpentry to Instructors Report

Prepared for

Software Carpentry Foundation and
Data Carpentry Foundation

by

Beth M. Duckles, Ph.D.
Research Assistant Professor
Portland State University
Portland, Oregon
beth.duckles.com
beth@duckles.com

January 2016

Value of SWC to Instructors Report

Table of Contents

EXECUTIVE SUMMARY.....	5
QUESTION 1 - WHY DID YOU BECOME AN INSTRUCTOR?	7
CONNECTION TO AN EVENT/PERSON.....	7
ONLINE PRESENCE	8
SKILLS	8
PEDAGOGY.....	9
REASONS WHY	10
ORGANIZATIONS.....	10
MISCELLANEOUS	11
QUESTION 2 – HOW DOES THE INSTRUCTOR USE SWC/DC SKILLS?.....	12
SKILLS	12
TEACHING.....	13
FREQUENCY	14
MISCELLANEOUS	14
QUESTION 3 – PERSONAL AND PROFESSIONAL BENEFIT.....	15
COMMUNITY.....	15
TEACHING SKILLS	16
LEARNING SKILLS	17
MISSION.....	19
PROFESSIONAL	20
NO BENEFIT.....	21
QUESTION 4 – STORIES OF SWC/DC’S IMPACT.....	22
STORIES ABOUT SWC	22
SKILLS	22
BENEFIT TO SCIENCE	23
SOCIAL/GROUPS.....	24
EFFICIENT	24
TRANSFORMATION.....	25
MATERIALS	25
EMPOWERED.....	25
FEEDBACK FROM STUDENTS	26
MISCELLANEOUS	26
NO RESPONSE.....	26
QUESTION 5 – SUGGESTIONS FOR IMPROVEMENT	28
ADMINISTRATION.....	28
ADVANCED WORKSHOPS.....	28
COMMUNITY	29
CREDENTIALS.....	29
DATA CARPENTRY	29
EMAIL.....	30
FOR PROFIT	30
GROWTH	30
INSTRUCTORS.....	31
INTERNATIONAL	32
LOCAL.....	32
MARKETING.....	32

Value of SWC to Instructors Report

MATERIAL	32
PROFESSIONAL DEVELOPMENT	34
NOVICES.....	34
RESEARCH	35
STUDENTS.....	35
TOO SHORT	35
TRAINING.....	36
RELATIONSHIP TO UNIVERSITIES	37
WORKSHOPS.....	37
MISCELLANEOUS	37
APPENDIX – METHODS.....	39

Executive Summary

The goal of this survey was to gather data on the value of Software Carpentry and Data Carpentry to our instructors as well as to find useful stories and narratives about how the organization has made a difference. To do this we chose open-ended questions to get a broad sense of the variety and range of different ways that volunteer instructors became interested in and derive benefit from Software Carpentry and Data Carpentry (abbreviated SWC/DC). See the Appendix A for more information on the methods used to collect and analyze the data.

Key Benefit of SWC/DC to Instructors

Three aspects were of the most benefit to Software Carpentry instructors.

- 1) Networking with a diverse community of scientists around the world
- 2) Learning how to teach and developing experience teaching
- 3) Teaching Software Carpentry skills to others.

Importance of Community

A key benefit of instructor's involvement with SWC/DC was the community. Respondents indicated that the community was beneficial for social connections, learning skills such as teaching or coding, strengthening cross-disciplinary networks and more. This is an important part of the reason instructors stay involved in the foundation's work.

SWC/DC Teaching Materials as an Asset

- Having the SWC/DC teaching materials on the web is an asset that should not be understated. Not only does the organization recruit instructors via the materials, but instructors also reported learning skills via the online materials and sharing the materials with others.
- Some discussed using online SWC materials and curriculum when they taught other seminars, classes and workshops besides those for SWC/DC. These seminars seemed to be of benefit to their organizations, labs and communities.
- The materials are a way that the SWC/DC community shares their mission. I encourage the community to see this as a significant asset.

Teacher Training

- There was wide positive regard for the teacher training workshop with many instructors indicating that this was useful not just for their ability to teach Software Carpentry but also for learning to communicate about science, give speeches and to teach classes beyond SWC.
- Some instructors also felt that this gave them marketable skills that they could use for their careers, giving them a credential for their resume, experience teaching or greater visibility in communities that were important to them.

SWC/DC Instructors are Passionate

- The enthusiasm and positive language used by many respondents was notable. They said things like “It’s fun!” and discussed their love of teaching, or passion for being a part of the community.
- Many of the respondents were able to indicate the relationship between the work they do as a teacher of SWC/DC and the work they do professionally.

Teaching Effectiveness and Student Follow Up

- There seemed to be a lack of clarity from some instructors on how best to evaluate their effectiveness as teachers and the desire for more data and research on the benefit of their teaching.
- When asked for anecdotes about their impact, some respondents said they didn’t have information or didn’t keep in touch with students after workshops. Some said they thought their teaching was effective but they didn’t know.
- Several commented that they would like to see more data gathered on this topic.

Room for Improvement

- While there was a lot of feedback for improvement, there is no overall agreement on specific changes that instructors would like to see. The enthusiastic response to the survey does suggest interest in improving and continuing the mission of Software Carpentry and Data Carpentry, but the specific steps ahead are not clear.

Organizational Mission

Drawing solely from the wording and descriptions of the work of SWC/DC via the instructors in this survey, the mission of the Software Carpentry Foundation appears to be:

To support reproducible science by teaching scientists the best practices for using open source analytic tools so that they can automate their workflows, collaborate more effectively and reproduce their work in a more efficient manner.

Though there are a variety of tools that individuals talked about using to accomplish this mission (git, R, Shell, Python etc), the underlying goal seems to be quite consistent.

Question 1 - Why did you become an Instructor?

The first question was to give us a sense for how instructors got into SWC/DC and what compelled them to become instructors. We asked: How did you get interested in doing Software Carpentry/Data Carpentry?

In this section, I group responses into several different themes with the number of respondents whose answer fit with this theme counted as an “n” (ie, n=2). Responses are not mutually exclusive. Very often responses are categorized in more than one code. The numbers following each quote are for identification.

Connection to an Event/Person

Having a connection to either a SWC event or to someone who was already engaged in doing SWC work was one of the top reasons that people became a SWC instructor. They discussed attending a workshop (n=36), helping out with a workshop (n=16) and even being a part of the process of hosting or trying to host a workshop (n=13) sometimes without participating or helping.

“By taking one [workshop] and realizing how useful it was for me.” (53)

“I was an assistant for a course, and just thought it seemed really fun” (58)

Some said they learned about SWC because of someone they already knew such as a colleague, mentor, professor or lab mate (n=36).

“My PhD supervisor recommended I improve my computing skills through SWC.” (124)

“A colleague of mine is an instructor and she needed some helpers for a workshop. Besides, my wife had attended a workshop and had found it quite useful. So I just joined out of curiosity and later on became an instructor.” (185)

“Someone in the lab told me.” (62)

Some were invited to be a part of the instructor program by someone they knew (n=22) or they named specific individuals who invited them such as Greg or Titus (n= 48). Some discussed being connected to a workshop by seeing it somewhere in their college or university (n=13)

“Greg Wilson called me and asked if our organization might be interested.” (134)

“My organization was interested in putting on Software Carpentry workshops and I was asked to get certified as an instructor.” (141)

“I had a class with SWC instructors before, and it was so impressive and interactive that I wanted to be one as well. After having the instructor's class with Greg Wilson and CO., I knew I wanted to keep doing it.” (136)

“There was a push to host a workshop at my university, so I signed up to be an instructor.” (8)

Research into nonprofit organizations suggests that this is not unusual. Volunteers are much more likely to participate when they are asked to participate personally by someone that they know or respect.

Online Presence

Respondents talked about learning about SWC based on finding the organization online (n=11), through twitter (n=11), through a mailing list or email (n=3), through a blog post (n=5) or a media appearance (n=1).

“I was discussing with colleagues the poor quality of software written by researchers, especially new postgrads, in our research group, and found Software Carpentry by doing a web search for approaches to training researchers in good software development practices.” (93)

“I heard about it initially on Twitter (probably around two-three years ago now) and thought it was a great idea.” (81)

“I found Software Carpentry through twitter and the blog. It felt like they were offering something I really needed at that stage in my career.” (110)

Several mentioned how important the teaching materials were in being attracted to teaching for SWC (n=13).

“When I was finding learning material, I found SWC. The teaching material is very detailed and self explanatory. After I read it, I thought I can get a deep understanding by teaching others.” (160)

“As a postdoc I was looking for best practices in doing computational science, so I found the lesson material on the web.” (17)

“I discovered the old Version 3 materials online as I was trying to improve computational practices in my lab. After reading and starting to implement them I knew that both I and science would benefit from me getting involved.” (70)

“I found Software Carpentry online while I was looking for resources to upskill myself. They were a wonderful, introduction to the collection of basic skills a scientist needs to get started, so I spread the word. I quickly became a person those wishing to learn programming came to for the 'Getting Started' spiel. The opportunity to become a SWC instructor arose and I jumped at the chance to get a 'teaching' related qualification' for my CV, but also to standardise my approach to teaching this kind of content.” (161)

Skills

The importance of the skills they were teaching were a significant part of the reason why some instructors wanted to teach. Respondents said that they saw a deep need for these skills and that SWC was a way for to help disseminate those skills (n =36).

“A colleague introduced me to the project. I had experienced first hand the need for software carpentry skills, and the difficulty of being self-taught on those skills in a field where they are necessary but there is no support for learning them. So I shared the value of teaching coding to researchers, and was excited by the opportunity SWC presented to pursue that goal.” (22)

“I realized that I needed to re-learn programming skills in a better and practical way. There are many useful libraries ready to use that solve that problem you have, but I didn't know how to use or modified them. Besides, I detected that many of my PhD colleagues were dealing with data but basically using Spreadsheets with no automation whatsoever. Our background using CS basic tools is sometimes limited as chemical engineers. Finally, I found SC and saw that Greg Wilson found this problems years ago so I tried to became part of it.” (155)

“Having moved from Physics to HPC, I know first hand the importance of the basic tools taught by SWC and am very keen to share this knowledge and experience with other scientists who use software in their research but who do not have an education in software development.” (180)

“As a mechanism to help our graduate students and researchers become more computationally literate, as a first stage in achieving fully reproducible research.” (41)

Some talked about the fact that they previously had had specific skills that led them to want to teach these things to people so that they also had these skills (n = 16).

“As a combination computer scientist/biologist, I regularly encounter situations where my computational skills allow me to do my work way faster than my colleagues without these skills. Since I got into computer science largely by dumb luck, I feel a responsibility to help pass on my knowledge to people who didn't have that good luck. We live in a time where our curricula haven't caught up with the importance of these skills. ... Not only does this serve people who don't happen to learn these skills badly, but it perpetuates inequalities due to the fact that people who are already disadvantaged (women, minorities) are the ones who are least likely to be exposed to these skills.” (34)

“I'm a faculty member who got tired of teaching these skills one-on-one to every new grad student” (39)

“I'd been concerned about helping graduate students improve their programming skills and so started a 1-credit university course; someone saw it and then pointed me to Software Carpentry, as an example of existing workshops with the same aim.” (65)

Some said that it helped to increase their skills and keep existing skills sharp (n = 5).

“I went to a SWC bootcamp and learned a ton. I wanted to gain more teaching experience and also learn more about good programming practices in science, so I became an instructor.” (35)

“I wanted to get better at the subject!” (48)

Pedagogy

Another significant rationale for being an instructor is the desire to learn, understand or be better at teaching and sharing information with others (n = 31).

“The way the teaching was done, collaboratively, very open and welcoming format, stickies for instant feedback, attentive to each participant's pace felt like a revelation for me. I knew I wanted to learn more about the teaching process as well.” (146)

“Some of the pedagogical and teaching tricks and best practices immediately improved short-courses and workshops I was already teaching.” (137)

Respondents talked about wanting to have more teaching experience or credentials to be able to share with others (n = 27).

“The opportunity to become a SWC instructor arose and I jumped at the chance to get a 'teaching' related qualification' for my CV, but also to standardise my approach to teaching this kind of content.” (161)

“I always felt I would enjoy teaching and I wanted to increase my skills in the topics. The instructor certification was important in making me feel more confident that I could do this” (75)

Some also talked about enjoying teaching and wanting to find places where they could teach what they like teaching (n = 6).

“I took a SWC in 2013 and fell in love with the idea of teaching people better coding practices.” (108)

Reasons Why

Some respondents also mentioned why they were so interested in being involved as instructors for SWC. Among the motivation for being involved were a sense of enthusiasm or excitement for the work (n = 8) or curiosity (n = 3).

“A labmate hosted a workshop, and I was a helper. I had a lot of fun!” (7)

“I volunteered to be a helper once. I thought it would be interesting” (187)

Several respondents mentioned that their personal goals were well aligned with the way that SWC conducts their work (n = 13) or that they really liked the SWC/DC community (n=2)

“Its core values, goals, and pedagogy match my own personal views and practices. When a friend pointed me to an early SWC site, I was instantly hooked.” (121)

“The gap between skills trained and skills needed in doing science is enormous; participating in SWC is part of my personal fight against this problem.” (51)

Organizations

Organizations and conferences were also mentioned as being places that people learn about SWC, events where a person presented about SWC/DC or where they learned that there was an option to become an instructor. The organizations respondents mentioned are listed below:

- ComputeCanada
- Mozilla
- Hacker Within
- SciPy
- EPCC/SSI/ARCHER
- Coursera
- eScience
- ResBaz
- Bioinformatic Workshop
- NGS
- SSI
- PythonMontreal

Value of SWC to Instructors Report

- ResearchPlatforms
- PyCon
- LBL
- Google
- eResearch
- Sesync
- WISE - Women in Science and Engineering
- OKC.js

Miscellaneous

Miscellaneous responses include doing SWC/DC because of the workplace they were hired into (n=8). One said they liked to travel, another was a founder of the organization and another respondent said they were not interested.

Question 2 – How does the Instructor use SWC/DC Skills?

The second question asked respondents to discuss the skills and uses of SWC/DC in their daily work. We asked: How do you use the skills you teach in Software Carpentry/Data Carpentry in your daily work?

Skills

Instructors specifically mentioned using many of the skills they teach in Software Carpentry and Data Carpentry in their work. Respondents talked about using Git, Github and version control (n=78), Python (n=59), the shell (n=54), R (n=28), SQL (n=9), C/C++ (n=4).

“I use the bash shell and Unix daily for running things. I use Python and git daily to write software and track changes. I manage databases, both a smallish SQL database and a large MongoDB one. I often perform quick analyses using Python in a Jupyter notebook, and I'll often return to notebooks and rerun cells with updated external data.” (139)

“Mostly Python for large-scale data analysis.” (181)

“I work as a training coordinator and as a lecturer, so I write tutorial materials for students from freshmen to graduate students, and I eat my own dog food using Git and scripting to manage everything in the pipeline.” (21)

“I use R extensively, every day, and the shell to a lesser degree. I try to practice the open science principles that SWC/DC stand for by making my work reproducible.” (174)

“Last night, my boyfriend looked at my laptop screen, saw the ocean of text editor windows, and asked me how many hours a day I spend looking at a text editor. We figured it was at least four hours on average. As a half computer scientist, I spend a lot of time writing code (mostly Python, but some R and some C++). Some of this is data analysis scripts, and some is maintaining the computational research platform that I use for my research. I keep it under version control and collaborate with others using Git. I do most of my work from the command-line, especially when I'm dealing with large numbers of data files on my university's high-performance computer cluster.” (34)

Some respondents said they use everything that SWC/DC carpentry teaches and they use it all the time.” (n=23).

“I can't really answer that question, because pretty much everything I do relates to these skills in some way.” (169)

“I write scientific software. In that process, I use every practice we teach.” (23)

“Completely thoroughly, for basically everything I do that is computer-related.” (65)

“Everything I do is managed in git, written in python, run in bash.” (85)

They use these tools to write code (n=36), do data analysis (n=21), automate their research tasks (n=18), do research (n=16), write scripts (n=14) and change or alter their workflow (n=10) or make the work they do more reproducible (n=8).

“I've tuned some of my own research workflow.” (171)

“I produce various scripts for myself or my clients from time to time. Occasionally I do complex data analysis tasks (of administrative importance, rather than scientific).” (15)

“I use version control and modular code on a daily basis. I am starting to use full automation of projects (from data to manuscript, figures and tables) in order to achieve full transparency and reproducibility.” (108)

Additional skills they use in their work include: learning testing (n=6), data organization skills (n=6), data visualization (n=4) and plotting (n=2).

“All of my physics simulations and models are written in Python. Additionally, literally all of my work and personal computer files are organized in git repositories. I evangelize for both (and Software Carpentry) frequently.” (152)

“I use R daily for making graphs and data vizualization” (10)

“I program daily in my research, so I use most of the SWC/DC curriculum every day” (44)

They also discussed make (n=3), modeling (n=2), statistics (n=1), high performance computing (n=1), programming (n=1), email (n=1), compiling code (n=1) and code review (n=1).

“I do computational disease modeling and program for maybe 70% of my job. SWC skills are foundational.” (122)

“Since I've left academic research, I unfortunately use them less. However, when reviewing code of junior developers of my team, I think SWC gave me lots of tools to explain a particular issue” (186)

Teaching

The second set of skills that respondents referenced were the teaching skills they learned in SWC Instructor Training (n=61) echoing responses in other questions. Indicated that this helped them teach about technical topics.

“I develop data lessons and teach workshops. I use the teaching skills daily.” (125)

“I am a professor and use various pedagogical approaches that I've learned in SWC in my work. I also utilize computational driven teaching that has influences from SWC.” (175)

“Someone said that the best way to cristalize, learn something for good is through teaching. Software/Data carpentry allows me to be more confident and comfortable with the skills (I think I) possess as an information technology professional.” (52)

“It is as fundamental as reading and writing. And it is probably because I am so familiar with computing that I had so much difficulty teaching novices. This is where the SWC material is so important - I use it as a roadmap for trading the students so I don't jump too far ahead without covering material I take for granted.” (135)

“Teacher training helped me to better understand barriers to learning and try to build that into my lesson plans. I structured my own intro to R course along Software Carpentry lines.” (187)

“ Major impact on how I think about teaching.” (111)

Several mentioned that they believed this helped them to communicate more effectively (n=10), teach their classes (n=2) and learn more effectively (n=1).

“I work as a PhD researcher in the area of Bioinformatics, which requires repeated tasks on different datasets and constant communications with the biologists. During the trainer's program I learnt various teaching skills (e.g. concept map) that helped me to convey my ideas more effectively to the other scientists who are not informaticians. Moreover, by teaching software skills to the scientists to a completely unrelated fields taught me to think about my scientific work from a broader perspective.” (166)

“I very much like the direct hands-on approach as a contrast to the traditional frontal lecture approach.” (178)

Frequency

Respondents mentioned that they used these skills daily (n=47) and one said they used the skills weekly. Many also discussed how foundational these skills were.

“I use them all day, every day, primarily to manage and analyze data.” (22)

“As a Research Software Engineer, Software Carpentry skills are the fundamental building blocks of almost everything I do. I would not employ anyone as an RSE who didn't have these skills.” (92)

Miscellaneous

Some discussed the benefit of knowing best practices in the industry because of being a SWC/DC instructor (n=7). Some discussed the benefit of these tools for writing (usually in regard to version control) (n=2). Two respondents discussed SWC/DC as being good for getting a job (n=2).

“Honestly I think the fact that I've worked with SWC has been very helpful on job applications for academic faculty jobs.” (58)

Respondents indicated they learned new tools from doing this work (n=2), one respondent indicated that these skills helped them to unlearn bad behavior. Some also say they do not use these skills in their work (n=5)

“Unfortunately, I primarily use email, Google Docs and M\$ Word in my work so I don't use any of the skills I teach in my daily work.” (97)

Question 3 – Personal and Professional Benefit

In the third question, we asked respondents: What is the benefit (personally and/or professionally) to you in being involved with the Software Carpentry/Data Carpentry Community? Almost all answers to this question had multiple codes.

Community

The most common response to the benefit of being a SWC/DC Instructor was that instructors liked that they could network and talk with people, make new connections and meet people who were “like me” through Software Carpentry/Data Carpentry (n=70).

“I’ve met so many great people and friends through SWC. It’s also feels good to be part of a large group all working toward a common, worthwhile goal.” (33)

“Personally, it lets me meet very talented, interesting, well-rounded and friendly people. Usually we bond easily because we have similar backgrounds, life experiences, and causes dear to our hearts. Also, the very experience of teaching is a very rewarding one, even though it’s difficult and exhausting. Professionally, it expands my network. Many colleagues see me as a bridge between communities (academic / industry, scientific / startup, Canada / USA...)” (159)

“- Networking. Knowing people around the world with the same interests than me is a great thing. SwC community is amazing, very helpful and full of resources.”(73)

“I’ve view my participation in these organizations (through mentoring subcommittee meetings, teaching workshops, twitter, etc) as an intellectual lifeline of sorts. The management and development of policies for these groups really model the type of transparency, clarity, and cooperation I wish was more pervasive in science. The attention to diversity and inclusivity help me feel welcome, and encouragement from others in the group let me know my contributions are meaningful and helpful.” (118)

“Huge. Lots of contacts from a broad range of fields. Many new approaches tested by people looking at similar challenges. Constant questioning of the ways that the details are done. Reinforcement of the many ways that a topic can be approached. The enthusiasm!” (111)

Respondents discussed the connections they made and the benefit in terms of the community aspects of Software Carpentry (n=35).

“The community is amazing, and it feels like a big family - for all of us scattering around the globe, it has been a bonding experience as well.” (136)

“The community - networking - and support solving informatics issues is invaluable! And it is very positive and full of energy.” (125)

Several responded that they liked to learn from other instructors (n=13) and that they enjoyed the chance to travel (n=10) and meet students (n=7).

“Meet interesting people that can lead to novel ideas” (62)

“I have gotten an opportunity to meet others interested in how the tools we use influence the productivity and quality of our research. “ (66)

“I get the chance to get to know the skill levels of researchers and their computing requirements for research, which enable us to develop training strategy and HPC training material.” (83)

Some responded that they thought Software Carpentry had a positive brand in the scientific community and that the name alone was worthwhile in conveying their interest in the mission of reproducible science (n=6).

“The community is a huge benefit. I also find it beneficial to be associated with a project as successful and with such impact as Software Carpentry.” (153)

Teaching Skills

Respondents said that they benefitted from learning better teaching skills and from learning to teach more effectively (n=46).

“I became a better teacher, get a lot of satisfaction out of giving workshops, have incorporated SWC teaching techniques in the other teaching I do, I feel part of a larger community of like-minded individuals.” (110)

“I appreciate the teaching about teaching that I received through SWC and I think I am able to teach beginners much better than I used to.” (57)

“I have re-branded myself at work from someone known solely as an educator to someone known to have programming skills who also teaches. This is very important as I transition from the (disappearing) grant funded position I'm on now to some other position TBD. At my workplace programming skills are few and far between, and the computer science core is sorely overloaded. There is recognition that it is better to "teach a man to fish" rather than to "give a man a fish." I've been recognized as someone who can teach others how to fish, which will reduce the computing core's workload and will make researchers more independent.” (104)

Some commented specifically that they really enjoyed teaching and benefitted because of the pleasure they got from getting to teach the topic (n=28).

“I love teaching, and I want coding to be accessible to as many people as possible.” (114)

“I am a very communicative person and I like to work directly with people. The opportunity to use my soft skills and passion to share scientific ideas with wider audience is normally not possible in an academic setup. However, working as a part of Software Carpentry foundation I developed confidence to combine both my passion for technical topics with my social and communication skills. Professionally it helped me to learn to convey my work to the collaborators in a more simpler manner and teach them basic informatics skills so that they can work in a reproducible manner.” (166)

“Feeling good. After doing a workshop and see how happy the students are from what they learnt makes me feel amazingly well. Tired too! but the amazing feeling wipes it away.” (73)

“Personally, I just find SWC teaching to be personally rewarding. It feels good to help people understand something that they were previously struggling with.” (64)

Some said that they got teaching experience for job applications or a resume/CV (n=22).

“As a graduate student, we have few opportunity to increase our teaching experience, especially in fields we are passionate about. It gives me both the confidence to teach, as was as important avenue for feedback about my teaching approach.” (205)

“For a resume, it demonstrates strong communication skills and ability to speak before audiences. On the job, it does help to have gone through the training, especially as it has supplied me with scientifically credible teaching methodologies and ways of thinking about communicating complicated technical concepts.” (126)

“I love teaching and I'd like to do that for a living someday after I finish my PhD. Being a SWC instructor provides me with the necessary experience I need to get such a job, and to learn some teaching skills myself.” (74)

Several also said that by teaching the material, they were more likely to learn it effectively (n=10) and one respondent indicated that they enjoyed teaching without having to grade.

“I always learn something new when I prepare the courses.” (128)

“I learn something every time I teach, plus getting to know the people who are part of this community helps since I now know people who can help me when I get stuck on coding something” (116)

“Taking time to try and explain techniques that you use almost every day help deepen one's understanding or extend. In some cases one is forced out of one's comfort zone to teach stuff that one is not so familiar with, e.g. python.” (28)

Learning Skills

Besides teaching, respondents said they learned other skills that were new to them, or refreshed existing skills when they taught (n=31).

“ I think I'm a better programmer, developer, and data analyst because of the reproducible philosophy SWC instilled in me.” (138)

“It has made us a lot better at the computational side of our research and helped us build a reputation as a good place to do computational research in ecology. It lets us tackle problems we couldn't accomplish otherwise and makes us more efficient.” (70)

“The use of github pages and former use of pandoc for lecture notes has taught me skills I would not have picked up otherwise. I now use github pages for my wedding website and pandoc to convert my latex papers to word for people that prefer to use words track changes feature to provide me with comments.” (130)

“Professionally, the main benefit is that codes and commands come to my mind so much faster that it's absolutely worth the time to dedicate to this great project!” (136)

Some said that they liked sharing skills with others and felt that they benefitted from that experience (n=21).

“Personally I benefit from the satisfaction of contributing to a wide variety of research projects with the skills I share with other researchers.” (180)

“It gives me an avenue for sharing the knowledge and skills I've gained during my post-graduate training. It also allows me to come in contact with a broad spectrum of research applications, which scratches my curious itch. There is no replacement for interaction with people who know more than you about things you find interesting.” (193)

“Within a research environment it is a professional benefit to demonstrate that you're helping others and sharing knowledge.” (200)

“Personally, it helps me not lose sight of fundamental issues that scientists who are novice programmers encounter. I help develop and maintain services for dissemination of data to scientists via a web site / apps, an API, and raw db access. It's crucial that these services be approachable and usable by people who are not professional programmers but that are motivated to learn best practices for efficiently getting at and analyzing the data.” (139)

The ability to communicate and present more effectively was another skill that respondents commented on (n=11).

“I think that being an SWC instructor makes me a better communicator. As an example, I recently gave a talk for one of our department's journal clubs. To prepare, I used concept mapping and other pedagogical techniques that I learned in the instructor course. I know that it helped my presentation. So one big professional benefit is that I am better able to communicate my work to my colleagues (and hopefully to the public too).” (64)

“Preparing for the workshops is a great review of the fundamentals, and working as an instructor has also helped me to be better at communicating about technology.” (141)

“I have received enormous benefit from being in the Software Carpentry community. I have learned a lot about programming from my fellow instructors (in person and via twitter). I have also learned a lot about how to be an effective teacher and communicator. Finally, I have felt empowered as a woman to take on leadership roles related to programming i.e. since joining SWC I have become a co-organizer for my local useR meetup group.” (150)

Respondents indicated that the materials and lesson plans were beneficial (n=10).

“...the public availability of lesson materials allows me to introduce a broad range of folks to these methods, either through inviting to workshops, or to work through on their own.” (118)

“I recommend SC lessons almost every week. There are many PhD students who did not learn (or did not know that they even exist) some of the tools SC is teaching.” (155)

“Access to well structured lesson plans that cover the material my colleagues in our research group need to know, and participation in discussion with well informed people with similar aims at other institutions.”(93)

“Crowd sourcing ideas that come together as great open source courses. It has gone through so many more iterations than anything I could develop locally.” (135)

Respondents also talked about learning best practices (n=8), the fundamentals (n=7), how to make their work easier (n=3) and how to write code (n=8).

“Increased knowledge in programming, best practices and open science. Become more comfortable with my skill set and sharing it with others.” (143)

“Filling a the important gap in computational literacy at the University. Writing scripts, using centralized code repositories and harnessing the command lines are a must for todays scientists. The Software Carpentry has build a well tested, through system for teaching those skills.” (71)

“Preparing for the workshops is a great review of the fundamentals, and working is an instructor has also helped me to be better at communicating about technology.” (141)

Mission

Several discussed mission related topics in discussing the benefits they gained from doing this work. For instance, some suggested that their work as SWC Instructors was an effective way to help science (n=21).

“It's a good way for me to learn more about best practices in computing, and to share that knowledge with other people in the scientific world. This makes me feel like I'm doing what science is supposed to be all about - sharing ideas and teaching others.” (101)

“The benefit to me being involved with the Software Carpentry/Data Carpentry Community is to continuously learn how to do sound science, follow the best practices in scientific computing, and meet and learn from other (geeks like me) scientists doing similar work. Other benefits include keeping up to date with the latest best practices, technology, and methodologies of doing sound scientific research.” (32)

Others talked more generally about the work being personally beneficial, rewarding and even fun (n=20)

“On the personal level - I actually started believing that I'm not too stupid for Computer Science.” (97)

“Personally, I derive a lot of self-esteem from Software Carpentry as I value interacting with learners and the challenge of relating technical knowledge in a way that helps people to learn.” (147)

“Personally, it's really fun and I enjoy meeting new instructors all over the place.” (35)

“Personally: I just enjoy teaching others awesome things.” (89)

Finally others discussed their admiration for the mission of the organization and their interest in being a part of that mission (n=16).

“I gain satisfaction from teaching and I believe in the SWC vision.” (30)

“Gives me satisfaction knowing I am helping researchers do better science. I am interested in working for companies that care about best practices, learning and growth, and this improves my resume in ways that would attract this type of employer.” (67)

“I’ve met so many great people and friends through SWC. It’s also feels good to be part of a large group all working toward a common, worthwhile goal.” (33)

“Professionally, it demonstrates that I have the skills that I teach and gives me teaching experience. I feel like it’s also a good way to signal my values (I believe in the importance of open and reproducible research) to others.” (34)

Professional

Some indicated that being a Software Carpentry Instructor looks good on a resume (n=19) and that it offered them job opportunities (n=18).

“I love the community, and while it hasn’t helped me professionally directly, it’s been on my CV and employers have liked seeing it. Also I see job postings from time to time so I think that it could help directly with job leads.” (197)

“My connections in SWC got me my current position.” (112)

“Personally, I enjoy teaching, and workshops give me a chance to travel places and meet new people. Professionally, I see a HUGE need for SWC/DC type skills in my field (genomics), and I feel an obligation to help others in my field when I can, even if that means sacrificing a bit of personal time. Workshops are also great professional networking opportunities. And while being a SWC-certified instructor may not stand out in a big way on a resume/CV, it could make the difference between two otherwise equally qualified job candidates” (100)

Others talked more generally about the professional visibility they gained by being instructors and/or hosting or trying to work with others to create SWC/DC events (n=17)

“Because of my involvement with software carpentry I have been able to host or participate in a number of events which I believe has helped to raise my academic profile.” (85)

“...whatever reputation I have among scientists stems almost entirely from my work with Software Carpentry.” (123)

“I think the work SC/DC is doing is vital and the materials are top-notch. I want to be associated with SC/DC because, while I haven’t contributed a whole lot since finishing my instructors training, I’m constantly on the lookout to incorporating SC/DC into the scientific computing efforts at my school (for instance, I think it’ll be likely I’ll get permission from my school to host a bootcamp in the next year). And SC/DC is such a cool group :).” (209)

Several talked about the cross disciplinary aspects of SWC/DC as a benefit to them (n=11).

“As a social scientist I don't often have a chance to interact with more computationally intensive scientists. Being involved with SWC gives me a chance to learn about the programming and data analysis habits and methods of physics, ecology, biology, etc. This gives me a lot of ideas and inspiration for my research, and helps me to feel more confident in my own computational work.” (174)

Others discussed the benefits in terms of the ability to collaborate more effectively (n=5) and the exposure they got to topics they wouldn't otherwise know of (n=4) .

“Opportunity to collaborate (e.g. on published papers)” (79)

“Running carpentry courses is useful to me since it helps introduce me to computational scientists on campus -- leading to future collaborations.” (92)

“Other benefits include keeping up to date with the latest best practices, technology, and methodologies of doing sound scientific research.” (32)

No Benefit

Eight respondents indicated that they didn't know of or had seen no benefit from being a SWC/DC instructor. One indicated negative benefits from being an instructor.

“Frankly speaking, I didn't get too much from SWC. I thought it mainly because in China there are not too much bootcamp here. But in future, I believe there will be more bootcamp.” (160)

“I'm still grappling with how SC/DC is benefiting my professional career. I've had people tell me to cut back in my teaching of workshops because it is not looked favorably by tenure and promotion committees. I don't really see a tangible benefit to my professional career, but I would like to see some efforts by the community in this area to recognize this conundrum.” (10)

Question 4 – Stories of SWC/DC’s Impact

We asked: What examples do you have of how the Software Carpentry/Data Carpentry Community have benefited others (students, labs, university groups)?

Stories about SWC

Respondents gave general comments about the benefit of SWC/DC (n=17)

“One student said 'This is the most useful course I have ever attended as part of my PhD studies'” (92)

“I think Software Carpentry gives scientists a vocabulary to google and to ask other programmers for help.” (91)

“Whether it's SWC itself or similar materials, the story is always the same. Some of the attendees come because they are obligated to, some come because they're curious, and some come because they are really motivated to learn and use the stuff. The first two groups are always a grab bag, but the last group ALWAYS comes away optimistic and grateful, well-equipped to build on the skills we introduced and use them in their research.” (100)

“One of the professors who attended a workshop here left the workshop to go immediately to work on a graphic for a publication. I taught a lesson on ggplot for publication quality graphics (my own lesson, not SWC or DC), and she put the knowledge to use right away. The ggplot graphics lesson was preceded by a full day of SWC training in R.” (104)

“After a workshop, I had an attendee tell me that while I was teaching the shell, he thought he would never use this EVER! The day after the workshop, him and his PI realized that this project would only be accomplished using the performance computing cluster which relies on the shell! There are also numerous examples of being saved by version control.” (116)

“Through running the workshops, we have helped so many people to see that scientific computing isn't something for geeks only. I've seen many many eyes light up with the realisation that they "got it".” (207)

“The D3 course I developed for SWC and taught once to 5 people so far (two larger courses coming up this month) inspired two students to go off and write amazing data visualisation websites.” (89)

Skills

Skills were an important component to the benefit of Software Carpentry and Data Carpentry.” (n=28).

“All of the data at our neuroimaging center is stored and analyzed on linux machines, yet there is no training on how to use the shell. We ran a SWC workshop for researchers and students at the center, and they were able to take the skills they learned back to their labs and begin automating their data management and analysis protocols.” (22)

“Some of my friends have used SWC as a springboard to gain skills necessary to transition into industry (especially data science). I've also seen students "hit the ground running" in research after a SWC bootcamp.” (114)

“Rapid bootstrapping of users regarding programming concepts. I wouldn't have been able to complete my thesis as quickly without this training---the face-to-face nature allowed it to fill in the gaps left by other courses.” (119)

“Now the people coming to ask me R questions (after our R workshop) can ask them in a way that makes sense, and I'm not longer spending all my time helping people load data into R.” (143)

“It is critically important to teach students with biology background computational knowledge and skills. They can benefit their research a lot. More importantly, if they don't want to work in biology research in the future, such experience in computation can also benefit a lot for their transition to other field.” (206)

Benefit to Science

Some commented on the benefit to their lab group (n=22) or the benefit to science generally (n=7) or that it helped with research (n=7).

“Besides the obvious advantages of using more agile and powerful tools, the software carpentry approach awakens on researchers the curiosity inherent of open source. If you have just found these new ideas and tools, what others might discover? In my particular case, working relationship with my thesis students it has greatly improved. Reproducible research tools used in thesis document edition are very useful, just as in the joint writing of scientific papers.” (113)

“Because of my involvement with Software Carpentry graduate students in my lab have taken and helped at Software Carpentry workshops and now use these skills in their research. My involvement with SWC also led to me and several other graduate students and postdocs initiating a study group at our university where peers teach skills they know to other peers.” (45)

“In our lab we moved from a hodge-podge of scripts or programs in Excel, Igor, C++, C, Java combined with a large dose of manual labour to writing all new analysis tools in Python. We have some experimentalist who have no programming experience and they are finally able to participate in the development of tools for data analysis and visualization. We use github and internal git repositories to share things we are developing. We base everything on Fedora, which gives us a uniform environment and up-to-date software.” (57)

“One student of one of the workshop I taught had to give a seminar on how to process some of their samples at Marine Biological Laboratory. She was able to learn and do a successful demo the week after she took the class. I was also a student. Without attending the workshop, I would've never or it would take me years to be where I am right now.” (68)

“It's mostly benefited my postgraduate students. I insist they do a SWC workshop, and then we run all project through github. They see the benefit, and go on to be much better researchers, both in terms of faster analysis, but also reproducibility.” (179)

“My graduate students all attend SWC workshops. It helps them manage the demands I place on them to work reproducibly. It has also helped to create a community across departments at my university for people who use open source software for research.” (174)

“I have seen entire labs gradually change their way of doing things, which has led to significantly more reproducible research.” (215)

Social/Groups

Several mentioned that groups of people benefitted (n=6) or there were social/community benefits from SWC/DC (n=4). Some also commented that SWC had a positive benefit with regard to getting work or jobs (n=8).

“Students we have helped at Hacky Hour following workshops have developed really interesting research projects that they'd not have known were possible if not for our course. It's also offered geeks a great place to relax and socialise with one another.” (38)

“SWC's greatest strength is its ability to catalyze the local community around a common need. From these workshops have sprung users groups and collaborations, as well as participation in the broader open science community.” (51)

“When people from a specific department attend, it helps them form networks with other people around them and they can help each other later. It is also beneficial for people who come along as helpers to build their skills and confidence.” (198)

“We have held two R hackathons after our first two workshops, and plan to hold another one on November 12. These are open to anyone who would like to practice their newfound R skills by helping on a project or asking others for help on a project. This has brought people together who otherwise would not have worked together to solve a problem. It's building a programming community here slowly but surely.” (104)

“Many people have told me how this training has allowed them to tackle new challenges; a couple of dozen have found jobs through contacts made via Software Carpentry.” (123)

“Well, for the students of my program, half of them now work in good paying tech jobs they didn't have before.” (140)

Efficient

Many said the tools allowed those they taught to become more efficient at their work (n=13), or that it influenced their student's workflow (n=2)

“The group I'm working in is more efficient. I didn't know anything about Software Carpentry before I started working with this group. Now we have multiple contributors to the same project. Our code is managed on a bitbucket repository and we share notebooks through nbviewer. It is a really fun and dynamic way to interact!” (19)

“One great example is a student who had to run 1000s of different versions of a mark-recapture model. He had planned to run each one manually through a command line program and had planned on spending the better part of a year's worth of evenings doing so. After learning these tools he automated the entire process saving himself 100s of hours of tedious work.” (70)

“What I live for when teaching is those moments when I teach skills that the scientist can immediately see will improve their pain-points in their current workflows. I remember a student who realized what a loop was doing and the expression on his face was like being given wings he could fly with.” (137)

“Using the tools and software taught by software carpentry, I've seen students at CalPoly SLO accomplish much more than they normally are capable of in school and personal projects. For

example, one student, using git and the Jupyter Notebook, built a self guided drone which had a rPi on it's back running the Jupyter notebook.” (149)

Transformation

Some shared stories of transformation through learning the skills of SWC/DC (n=10).

“At every workshop I have taught at, I see the students go through a transformation from the start of day one to the end of day two. Students start a little nervous and unsure of how computational skills will directly benefit their work to finishing having learned so many new skills, having much more confidence, and so eager and excited to directly apply the skills they learned.” (32)

“One example: thanks to our workshop in Kraków, one 19 years old student from high school will probably start a career in Computer Science. We'll follow up on him in a few months.” (63)

“All my teaching UG teaching. I teach 90 students per year. I went from having a course that was universally hated to one that is one of the most popular! Students are increasingly using more programming now in their course work and I have been asked to mould other problem courses in the same fashion. I even have other lecturers coming to my lectures to observe because they want to reproduce the success.” (135)

“SWC opens up the doors to a computing environment that most weren't aware existed. I've seen colleagues discover the command line, programming and version control that otherwise probably wouldn't have with alter comforts of a workshop.” (186)

Materials

Several commented that the materials were beneficial (n=9)

“Apart from the students learning in the workshop, I recommend the software carpentry lecture notes to all students working in my lab. Although they have not attended a workshop, they have picked up the skills by working through the notes.” (130)

“I often point people in my lab and other colleagues to the SWC lesson materials for reference, and encourage everyone to attend a workshop in person if possible.” (2)

“I've relied heavily on the lessons developed by these groups when teaching both undergraduate and graduate level bioinformatics courses. I can safely say that Software and Data Carpentry have been the most influential, driving force in developing pedagogical methods to teach programming to my biology students. I've encountered many students at workshops who had previously attempted to learn programming but had less than satisfactory results. The approach used by the Carpentry groups is the best I've seen at reaching a broad audience.” (118)

Empowered

Some talked about how SWC/DC empowered their students (n=7).

“It has motivated hundreds of instructors and thousands of students to learn more about computing in a scientific context.” (20)

“I have observed many instances how students in Software Carpentry courses feel empowered by their growing confidence. I appreciate seeing the students' increased interest and desire to learn more about the tools and methods taught in Software Carpentry courses.” (208)

“I know it gave a few people the confidence to start working through some of their programming problems on their own, rather than asking someone else to do it for them.” (8)

Feedback from Students

Some mentioned generally positive feedback (n=7)

“Feedback after the workshops (sometimes years after) from people that found them beneficial to their work.” (98)

“I just taught a Data Carpentry workshop at a Psychology department at another university. A few choice quotes from the student feedback that I've gotten so far: "The use of ipython notebooks made everything clear, easy and comfortable to look at, and nicely organized! I will definitely be using Jupyter in the future to share my codes!" "I look forward to more like this in the future!"” (169)

Miscellaneous

Some commented that SWC was a good introduction to the topics (n=3), others discussed that their teaching or communication became better because of teaching (n=3).

“The feedback from the students ranged from: "Wow, now I can make my own pictures." to "I can see the source code!? Really!?" That may sound weird, but to me it is very satisfy to empower a student to produce his own figures rather than counting on "that friends who know computers." The second comment maybe is even more important. The benefit that comes from getting students, that are using black box tools like Matlab, Statistica, etc, is priceless.” (43)

“I know many people who have used SWC as their introduction for using R for data analysis. This decreases the time required for analysis and moves it closer to the bench allowing for more flexible development.” (18)

“Our graduate students directly benefit from the workshops because they are given computational tools that they will use from day one in our program. They are introduced to version control early, rather than late (e.g. as the result of some versioning/backup disaster). Even though they will not master everything, they build off of the introduction that the workshop provides to establish a foundation of skills that develop according to the research needs of the individual.” (41)

“I think my lectures are better than before.” (77)

No Response

Finally, a significant number of respondents either had no data to report (n=30) or did not answer the question (n=35). Some commented their knowledge was anecdotal (n=5) or indirect (n=3). Finally, there was a trend among many folks that the information on outcomes from their teaching was simply not available. This suggests more research to gain concrete data would be beneficial.

“Only hunches. Lacking long-term followup, I'm always unsure how well anything sticks.” (24)

Value of SWC to Instructors Report

“No very specific examples come to mind. There is however very positive general feedback. Unfortunately, time does not allow for long-term feedback for impact assessment.” (180)

“I wish i had more of these! I know people use the skills but i don't have concrete examples.” (125)

“Hmm, I don't have clear examples. I guess I just have the impression that workshops do indeed help raise the mean competence level wrt software skills at the institutions that host the workshops. I don't know this for a fact, though.” (139)

“Unfortunately I don't have any first-hand accounts; I assume I've helped those at the workshops I've instructed, but I don't know for sure!” (167)

Question 5 – Suggestions for Improvement

We asked: What suggestions do you have for improvement? All are direct quotes with no editing, some responses are split if there are separate ideas. Respondents who left the answer blank or wrote responses such as “no” or “none at this time” are not included (n=78).

This section is organized coded differently. Codes indicate the general topic of the suggestion for improvement since there was not enough commonality among suggestions to further refine. Every answer we received is listed below.

Administration

- “Reduce the amount of admin and organizational scaffolding. I am very skeptical that a \$2500 administration fee helps anyone but a few intermediaries that are not really needed in the first place. Administration in an open and free initiative such as SWC shows that some rot is starting to show. Lean and less is better, in my humble opinion.” (52)
- “The governance of SWC is still not 100% clear to me, especially who is in charge, who to email if you have questions, who has the authority to make certain decisions, etc. (88)
- “Be more organized - I was never very good at that bit.” (123)
- “Obviously, more instructors and more admin support. Also a way to report back workshops held/to be held to the SWC/DC mothership so they can all be placed/advertised in a single location. More easily accessible instructors pool so as to find instructors more easily (I understand that this is part of what SWC/DC are charging for and that it may not be possible).(154)
- “It's pretty good, but obviously the appraisal system needs work - it just ridiculous that one man is the gate keeper. I also think the standards of teaching required to be a SWC instructor need some work. A bad experience with a poor teacher at SWC workshop can do harm, as it can actually turn people away from the very thing it is trying to teach.” (164)
- “Communication needs to improve, it's quite bad: I asked more than once about inclusion as instructor and never heard back from organizers and later found out on my own that the workshop was already completely staffed. That's not how you treat people. If you do not want to consider using someone as an instructor tell them and tell them why this is. This questionnaire definitely is a step into the right direction.” (177)
- “Can someone please explain reimbursement. What can we ask for? Airport parking, travel to airport, etc?” (31)

Advanced workshops

- “Find a way to get back to teaching more advanced workshops. I know the pressure to teach novices is strong, but one of the real strengths of Software Carpentry is teaching good self-taught programmers how to be much better.” (70)
- “While I realise the core focus here is the Software Carpentry workshop format, an area that needs improvement is follow up after workshops. I think you've done a fantastic job on the quality of the material and instructor training. More guidance or advice on how to continue engaging the community and help them use the skills covered in their own research would be really helpful. While the instructor training emphasis on novice audiences (which we often encounter in my biologically trained peers) is really helpful, the idea of intermediate level R or Python courses has been raised both in our local group and on SWC forums. This would allow more focused sessions for absolute beginners (time is often pressed here) and the more advanced session would appeal to those familiar with the programming languages. The would not only present an opportunity to show advanced users the latest cool packages but reinforce best practices and cover BASH/Git/SQL which many in scientific computing are not familiar with.” (84)

- “A lot of our users seem interested in an Intermediate level SWC type class (what happens after the first class). It might be worth scoping out a longer SWC class.” (134)
- “Expanding the list of topics we teach. Not being afraid to teach non-introductory topics. Teaching domain-specific tools.” (157)
- “Developing more advanced curriculum (currently we only have the novice lessons) (191)
- “Maybe.. in some time soon we should be looking towards a second level course, where it teaches an step further.” (73)

Community

- “This would be a big suggestion, but I would suggest that SWC have some sort of monthly newsletter or something specifically tailored to people that have taken a bootcamp but are not instructors. People often want follow up material after a novice bootcamp, but in many cases intermediate bootcamps aren't organized at their institution. What if we had a monthly newsletter or youtube video or blog post that would cover short blurbs of follow up material. For instance, maybe there would be short, 5 minute video tutorial on how to use some fancy python library that people may find useful, or a useful unix command that will find all the files that end in some extension, etc. This would also be a great place for people doing instructor training to test out material and teaching without bloating the repos and lessons. Also, keeping that constant line of connection with old attendees will probably lead to more bootcamps, as SWC will stay fresh in people's minds. It would be a TON of work though, so I completely understand if there isn't the desire to do something like this.” (35)
- “Improvement is difficult when everything is so good! So my first suggestion is don't leave it go. Keep the community engaged as it is while keep the enjoyment! I'm looking forward for the retreat, so probably I will suggest more of these.” (73)
- “More connection between instructors. As SWC has grown, I feel less connected to the other instructors, less aware of what is going on more generally, and unsure of when meetings are happening and which ones are for specific participants vs. for the group.” (79)
- “I haven't really done any actual Software Carpentry events yet. There aren't many happening and it's also hard to find the time. I wish there was some online version I could be plugged into on a monthly or bi-monthly basis just to stay in touch with the curriculum and the community.” (140)
- “More communication, more interaction between people, more fora for peer learning.” (215)
- “I was going to say that it would be good if there were more opportunities to interact with the aforementioned awesome community, but honestly there are an impressive number given how widely distributed across space the community is.” (34)

Credentials

- “That's a hard one :) Maybe SC can create some sort of "Certification Path" consisting of various Software Carpentry subjects, on various level of proficiency.” (25)
- “I hear the accreditation scheme is improving, which is good. I'm happy to help with expansion to Germany (where I'm working), and a friend wants to do the same in Copenhagen. Let's make it happen! (38)
- “A "tangible thing" to take away from the workshop? Maybe one of the digital badges as a qualification? Just something to remind them, when they're struggling with coding a month after the workshop, that they can do this...” (111)
- “I do like the (already in the works) idea of individually certifying people to teach each lesson. I feel like it would help improve individual instructors' confidence levels, and like it would reduce the variability in teaching quality that sometimes exists.” (34)

Data Carpentry

- “I am a big fan of the narrative arc that Data Carpentry workshops often have and would like to see something similar for SWC. This reduces the ability of the lessons to be completely modular, but makes teaching them (and motivating the progression of material and exercises) much more straightforward and helps students see how everything fits together. E.g. Start with dataset in

python/R, move to shell to automate, add version control, then back into python/R for exercise integrating all earlier content. A version of this that I have taught is here:

https://raw.githubusercontent.com/naupaka/2015-10-03-ua-iplant/gh-pages/workshop/01_Saturday/01_Saturday_am/SWC_workshop_overview.png.” (2)

- “Clarify the relationship between SWC and DC (I get it, but I think it's lost on many)” (46)
- “Maybe make Data Carpentry skills a prerequisite for Software Carpentry, so that the SWC workshops can focus more on best practices in software engineering aspects and eliminate some of the basic programming instruction.” (65)
- “Software Carpentry is a great, lean program. I think it works well as is. I am excited that Data Carpentry is starting up and running well as a more focused workshop. Other workshops or continuing education courses might be a good addition later on as a way for past Software Carpentry students to stay connected and expand their skill sets with focused workshops in topics like Social Network Analysis, Qualitative Data Encoding and Analysis, or Agent-Based Modeling.” (201)

Email

- “The email lists don't really work for me in their current format. Too many messages about lesson development that don't really interest me (just took a look at the list and don't really have many ideas for improvement).” (18)
- “As I said above, I haven't been very engaged recently. I do get a lot of email that are not relevant to me (e.g., looking for instructor for bootcamp in country X), and so I redirected all SWC to a special mailbox. And I don't look at this mailbox often, and probably miss more important emails. So, maybe being a little more targeted with emails...? I know this is hard given that people may want to travel to help. What about setting up an instance of Discourse? (<http://www.discourse.org>) We use this for quantitative advice at Macquarie. Great forum software.” (179)

For Profit

- “Perhaps offer a paid tier for delivery of workshops at for-profit companies or where it is being used for/organised by organisations that can afford to pay for it. It's one thing when we're helping grad students and ECRs be more efficient in their work, it's another when funded IT education & development programs are being wrapped up because there is "no need for teaching these skills outside the workplace", and yet the potential attendees are queuing up for - and encouraged by their supervisors - to attend the free SWC workshops.” (173)
- “There's still a need for SWC/DC in industry but it's a hard nut to crack.” (186)
- “I stopped volunteering after the decision to teach at for-profit corporations. I think it's exploitative to use volunteer labor for that.” (Email)

Growth

- “I see two areas which I think SC/DC that worry me: 1. I think there has been a tendency for SC (and DC to a lesser extent because it's just getting off the ground so to speak) to grow too fast -- I think there are a lot of instructors coming out of training without having any teaching experience or having very developed skills. I could also argue I'm not completely advanced, but I've taught with some newish instructors that aren't really at the level to teach yet. I wouldn't want to turn anyone away, but perhaps there should be more hoops (i.e. training) to jump through to be an instructor?” (10)
- “That's a tough one... I find that I am following Software Carpentry developments less and less, because its activity has exploded and keeping up to date would take too much time. I don't think there is a way to fix this, so I have no real suggestion. A related issue is the increasing bureaucratization of Software Carpentry, which reminds me uncomfortably of the paperwork in my day job. But again, I don't see how to avoid this. Growth is good, but has downsides as well.” (37)
- “If you look at the plots of SWC growth over the years (<http://software-carpentry.org/blog/2015/10/recent-stats.html>) The growth is linear. SWC and DC need to figure out how to make their growth exponential. This probably entails changing parts of the model.” (13)

Instructors

- “Obviously, more instructors and more admin support. Also a way to report back workshops held/to be held to the SWC/DC mothership so they can all be placed/advertised in a single location. More easily accessible instructors pool so as to find instructors more easily (I understand that this is part of what SWC/DC are charging for and that it may not be possible).(154)
- “Very interested to see how the instructor training day comes together. Also very interested in courses being offered by Titus Brown, I'd love to see more of these specifically targeted at skill development for instructors.” (18)
- “Be wary of overengineering. There's been a great deal of effort internally to build intricate pipelines of instructors, but I'm not convinced our current level of understanding of *why* we lose instructors supports the increasingly complicated procedure for becoming one. Before things become utterly baroque, we need to find some money to do some serious research on the psychology and sociology of the instructor experience, and use this to inform a simple and effective strategy for cultivating that experience.” (51)
- “A map of instructors with pins to help generate more community.” (62)
- “Vet teachers a little more... I co-taught with someone who was very off the cuff, and I had to reteach the lesson because students were really upset and confused. I personally used old lesson sets because I've never really liked dropping people into a bunch of data and starting from there. When I taught in Hawaii, I talked to the sponsors first, and I wanted to go with the new material, but after the first day, I switched back to the old material because I didn't feel like it was right for my particular set of students. I also think that some instructors spend more time on some things than others, and it's hard to figure out what you really need to get across and what you can drop if you're running short on time (I have a good sense of this bc I do software development in my day to day, but it's hard to have unless you teach the class a couple of times). I think there could be some instructor notes on what to drop. I suggested that people should meet up a month from the class and see what help they needed from the instructors that were in town and part of their labs.” (91)
- “So far I know, now the instructors training is currently in a transition time that involves new procedure which looks to me already improved in respect to the previous version.” (94)
- “Make instructor education not stop after the training: I think it would be very useful if instructors could get an occasional session revisiting some of the aspects taught during instructor training, to refresh and expand pedagogical skills. Ideally, instructors would pair up more across institutes to learn from each other. I, for example, would like to observe others teach. But logistics make this not always easy, I realise...” (110)
- “As new instructors we may have seen a workshop up close as helpers, but we are pretty much on our own running a workshop using our common sense mixed with some of what we learned. I think that reasonably new instructors could benefit from studying how more experienced instructors approach the task, perhaps on video.” (117)
- “When instructors are to lead an effort together, they should get organized at least a week beforehand (phone calls, shared material). I've noticed that when it does not happen, inconsistencies may happen and disputes/competition are more likely to permeate the class, which should be collaborative instead.” (136)
- “More in person training/experience/cross-pollination for instructors. I have not yet instructed at a workshop and although I would like to, I also feel a bit underprepared since I have not yet had much experience.” (144)
- “I think a major challenge of the organization is figuring out how to scale responsibly and sustainably. I regularly meet people who are interested in becoming instructors, but the wait-time is quite long. At the same time, as the organization grows, how do you ensure that instructors are getting adequate training, feedback and support. I think SWC is handling this challenge pretty well. I especially like the bi-monthly virtual sessions to discuss recent workshops. One thing I wish i had gotten in my instructor training is more examples of how people have taught lessons. However, i think in the new training there are videos of people teaching lessons, which i think is a great resource for new instructors.” (150)
- “I once had a workshop canceled because the other instructor could not come. As Software Carpentry grows, I think it will need to have a mechanism in place to solve this problem (for

instance, have a few people who can substitute for most/all core topics at anytime, with the last-minute tickets paid by Software Carpentry)." (214)

- "Volunteer-lead initiatives can rely heavily on people with more altruism than self-preservation. It is vital that participants (in particular students and postdocs) are not over-worked and under-paid (they get enough of that at home)." (216)
- "Have star evaluation system for instructions from students attend the workshop." (3)
- "I think we should focus on getting new instructors comfortable with all of the material and helping them lead their first workshop. As it currently stands, many newly minted instructors never teach a workshop." (23)
- "It would also be interesting to see what the return rate is. Oh it might also be worth considering that, to keep your instructor status, you must teach in at least one course in a year. I belong to the STEMNET Ambassador network in the UK and to keep your Ambassador status you must participate in at least one event per year. That would be an incentive to keep your instructors active (but then you could also lose a large part of them too)." (28)

International

- "SWC has reached a point where it is becoming increasingly an international exercise. A lot of SWC course material is developed for a western audience, with high level of exposure to technology and technological concepts. Improving the versatility of the course material in this regard might be a good time-investment." (193)
- "We also need to outreach to areas which are not UK, Australia, US and Canada." (97)

Local

- "I think there should be a more efficient mechanism for local interaction. People who want to run a SWC have no idea if there are local instructors. People interested in running have no idea if there are other SWC workshops already at their institution. This information is "out there" but it's not obvious, and this has causes confusion in my limited experience." (49)
- "More events in the NYC area! :D" (55)

Marketing

- "Maybe a stronger push for marketing / branding / etc. One of the biggest challenges of SC is convincing scientists that they need to learn good programming skills in the first place (or at least that it's worth the time in their busy schedules). I know a lot of people that think it sounds sort of interesting, but aren't sure whether it's worth it, and I think that's partially a branding/image thing. Focusing lectures / promotional material on making them better at data analysis will probably be a stronger message than improving reproducibility in their work or reducing the amount of time debugging." (101)
- "I think better promotional materials that sell the power of coding would be very useful to increase attendance." (142)
- "The visual branding (logo, etc.) is in need of improvement." (153)

Material

- "A markdown document with instructions on how to use github pages." (79)
- "Trainers develop their own style of teaching and own teaching materials specific for their audience. Software Carpentry foundation is not very strict about the teaching materials however the community is still trying to standardize the teaching resources. My suggestion in this direction would be to not make one kind of teaching material but allow trainers to contribute to different repositories of teaching materials based on their target audience. This might help future training courses by allowing us to pick the materials of choice." (166)
- "Foremost I need to get more experience with teaching. I may have some suggestions for the python part, which I'd fill on github." (185)

- “I would say there is too much material in most of the novice tutorials to be taught within a half day.” (4)
- “Would be great to find a way to better record what people are teaching (what changes are they making to existing lessons). In every workshop I've taught, instructors change the lessons but this is rarely recorded anywhere or examined in detail. I don't have a good suggestion about how to do this, though.” (6)
- “The teacher training with a focus on teaching theory and methods is fantastic, but I do also think a stronger focus on the materials (maybe concurrently) is also important. Would like to see more focus on providing opportunities for new instructors to learn the materials outside of bootcamps/workshops.(14)
- “The SWC github repository materials could use some reorganization. Particularly to new users, it is very confusing navigating them and obtaining /extracting what you need. The approach to teaching Git needs an overhaul. I have yet to see it be impactful to students in its current form.” (41)
- “Honestly, the shell and git stuff is a waste of most students' time. I get that they would ideally be using that stuff, but most of the students I've taught are not at any sort of programming level where that stuff has any value to them whatsoever. It's the medicine they have to take to get the education they actually want in R, and when they find out that it's half the course they're quite frequently disappointed.” (58)
- “The materials could be improved with a little more central control. Maintainers should push for their particular vision of what lessons should be. When these visions conflict, the two lessons can be compared objectively. Not every new idea can merged into the existing lessons.” (66)
- “just keep working to improve the lessons. think about how we can do add on or extension courses to get beyond the bare minimum.” (85)
- “In general, I think the lessons are great. More specifically, the Python lesson needs some work as after teaching it two times, I find I cannot get to the errors and exception part (and everything after that) within a 4 hours block. The pace seems just about right from what I gathered from the post-its and questions people ask in the training.” (96)
- “The infrastructure which the instructors use needs to be more stable. It changes way too fast and people who have time to teach twice a year feel overwhelmed as it turns out that the way they used to set up workshop website has changed 1001 times in the past 6 months.” (97)
- “I think the lessons are already going to be improved (for example, a NEXT button at the bottom of the lesson so that you go directly to the succeeding lesson). The instructor training is also being remodeled, so I'm more likely to have suggestions after seeing revised versions of both lessons and instructor training. (104)
- “I know there already is a bit of this in the curriculum (although every workshop I have been involved in did not get to it) is that we should really have some kind of capstone lesson at the end of the workshop where we use shell/R/github to accomplish some reproducible research, even if it's just forking a pre-made repo and making a few plots or something? (166)
- “More instructors and course content from other research areas.(119)
- “More stated power for lesson leaders (editors) to drive major improvements and make calls on pull requests. More retreats! More training on: how to contribute and how to teach the current materials. “More attention to making lessons teachable (tips for which parts can be left out versus those that absolutely should be included; other tips for areas of flexibility).” (120)
- “The group is headed that direction with the growing of the community, improving the teaching materials and reaching out, broadening the groups that can benefit by adding teaching materials for those groups. Ex. Librarians and Humanists.” (121)
- “I think some of the data carpentry lessons need more refinement. this is because they are young so i understand. but i've had to rebuild lessons before teaching before which i wasn't expecting. This was last year, however. it might be greatly improved now! :)” (125)
- “We need better curation of the GitHub lesson repositories; there's a lot of bugs in the Python and R lessons. The distributed nature of Software Carpentry scales really well but I personally find it hard to keep up with the e-mail listserve; it's hard to not feel isolated. We have a good "subcommittee" at the University of Michigan, however, that helps me feel connected to other Software Carpentry instructors, helpers, and learners. I think expanding the opportunities for learners, perhaps through

the development of an institutional model, e.g., a "code help cafe," could extend the relevance of Software Carpentry beyond a learner's first workshop." (147)

- "Python, shell and R lessons with this technology <http://mybinder.org/> (as a test)" (155)
- "Make it easier to improve the lessons, I have the feeling they are too slow to improve. For example I think we should switch from teaching numpy to teaching pandas, read a labeled column dataset instead of the inflammation dataset." (162)
- "One major suggestion is that the material needs to be cleaned up. That is, people contribute and the material expands - - but ppl fail to update anticipated timings. And so when you go to teach you find yourself overwhelmed and the pace is entirely too fast for students. A solution to this, is having lessons divided up into independent modules with accurate timings for each. Each module has prerequisite knowledge stated up front. This way based on the audience the instructors can selectively assemble the lesson to be taught over a tw-day period." (213)
- "I think we're doing well in general terms. I'd like to engage more people in lesson development: we have increased the number of instructors substantially in the last year, but the number of contributors have not followed the same trend." (217)

Professional Development

- "I also would like to see some initiative to acknowledge the efforts of community teaching in professional development. I want to commit more time to SC/DC, but I feel like this isn't respected in the greater scientific community. I know this isn't relegated to SC/DC in general (it's part of a wider issue on how training and teaching are respected in tenure and promotion). I wonder if SC/DC can really push the idea that instruction is important -- I think SC/DC is somewhat acting as a stopgap for the flood of people who aren't properly being trained via the traditional methods (University, etc.) and I think we need to be more vocal and/or proactive on the need that SC/DC serves and the importance of what we do." (10)

Novices

- "Have some sort of selection criteria of participants. Priority should go to novices. Last workshop I taught in had lots of green stickies all the time. Now this does sound like a good thing, but it was mainly because participants weren't novices and were rather intermediates looking for some solution in their research (e.g. 3D plotting I don't know what..). SWC needs to be for novices only, I think. Cz these intermediates may be taking the spot of someone who is a novice, and strongly needs suck skills." (74)
- "Sometimes people get lost and then they can't catch up. Some information on whether this is helping researchers longer term would be useful. it seems like they enjoy learning it but whether they use it is another story." (78)
- "SWC has become quite the big machine. It is likely intimidating for newcomers. I'd work on easy entry points for new instructors. Now that I'm a professor the incentives for teaching a SWC workshop aren't relatively as strong so it's hard for me to make a decision to sign up to teach. Plus my time is not as flexible and free as when I was a grad student and post doc. So it's not clear how to best fit in a the moment." (175)
- "The hardest thing is trying to gauge the experience level. There is always such a diversity. I think data carpentry should be really picking up some of the more novice courses. I often have ~30% wishing it was more advanced and then atleast 15% of who are struggling. Its a challenge to make that all work in two days." (205)
- "Our biggest challenge has been in figuring out how to accommodate the wide range of prior experience that people bring to the workshops. We certainly don't want to overwhelm the novices, who are the primary audience, but at the same time don't want to alienate the more experienced learners who have taken the time and shown the interest to be there. Splitting the course has proven to be a logistical headache. I don't know how to improve this situation, but I foresee it continuing to be a problem." (211)
- "I haven't taught in Data carpentry workshops yet. But from the Software carpentry workshop's experience, I think there is some scope of improvement in how we get to know the skills of participants prior to workshop which will help us in scaling our teaching material effectively and in

turn effect the way we communicate and teach these skills. Many times, we have a class with participants ranging from novice programmers to advanced. There comes a situation when either novice people lag behind during the course or advanced programmers feel that's too basic for them. Hence, a more intuitive way of getting the distribution of participants skills beforehand would help in catering needs of learners. Secondly, the design of teaching material to be well aligned with domain of research of attendees would give them hang of why, how and what they're learning and how they can use these skills in their research work which will also help them in connecting the dots together based on the skills taught during the course.” (212)

Research

- “Having participants state explicit expectations/results in structured pre-post workshop surveys ala The thing I'm most anxious about this workshop is _____. The one thing I hope I learn in this workshop is _____.; The most interesting thing I learned today was _____. The most useful thing I learned today was _____. The best thing about this workshop was _____. If I could change one thing about this workshop it would be _____.” (103)
- “I keep hearing about quantification and assessment of SWC products. Where Brooklyn at? Where are these assessments and data? Why all talk and no rock? (155)
- “We need a way to assess students and our impact.” (203)
- “Long term studies of course attendees would be useful - for a given course measure the impact of a course immediately after, 3 months, 6 months and then a year. Software Carpentry was supposed to go viral where attendees would be come helpers then instructors and so on ... that never happened.

Students

- “Long term followup, start doing what many seem loath to do: test our students. Combine summative and formative assessment so we have concrete examples of where we're failing, instead of relying on students to tell us their impression of where we're failing.” (24)
- “Trying to include time to include a capstone project that shows a nice clear example of how one would directly apply the skills learned.” (32)
- “I think the program is fantastic! My one suggestion would be to move away from teaching students how to use Office Libre (or any spreadsheet program) to work with data prior to jumping into R. I feel that one of the central goals of SW/D carpentry is to teach students how to get away from the limitations of spreadsheets, and that jumping straight into R is acceptable given this goal.” (152)
- “I think it is really important that learners feel part of a community. It is more likely that people will take what they've learned at Software Carpentry if they see examples of other people benefiting from these skills. I think there should be a focus setting up study groups to continue after the workshop.” (19)

Too Short

- “I have talked to Greg about this, but I have stopped teaching and hosting the two-day bootcamps because I believe they are too short to have a lasting impact, other than motivating students that they should learn this stuff. At the worst, they over promise and under-deliver in applicable skills, and this could actually discourage students -- giving them the impression that they are not cut out for the subject matter. One of the biggest problems in teaching computing is the intimidation that students feel, and it can take a week of "hand-holding" to get them over this. At the last two-week class we taught there was a student who really struggled. She had paid her own way into the class and drove many hours to get to the venue. After a rough start, by the end she was answering tricky questions with confidence and writing scripts, I gave her the informal "Most improved" award. She just wrote me recently thanking me for the confidence that the course gave her as she applied to grad programs. I think if she had been in a two-day class it would have blown her out of the water. I know not everyone (instructors or students) can spare that kind of time, but maybe could try a series of more focused one or two day courses covering a specific topic at a time would be feasible? Not sure!” (20)

- “I only taught one SWC so far, but the number only negative feedback I had from virtually all the students were the short duration of the Workshop and the fast pace. I can only reduce the pace if I throw content away. And teaching for more than 2 days is usually difficult. Sometime I think that the workshop model should be revised to encompass only one topic. I'd gladly teach a 2-days git course for example.” (43)
- “Some instructors need to slow down a little bit. They feel they need to cover everything in order to have a successful workshop. Sometimes less is more. It is not about covering every single section, it is about bringing people along with you.” (69)
- “I have said this a couple of times already (sorry for being insistent), but I do think that, whenever possible, instructors and organisers should consider a 3-day bootcamp. This way we can better intertwine theory with practice. I taught a Python bootcamp back in 2013 (my first one) where the third day required students to develop a small project, from data to some results using the full integration taught during the bootcamp (modular code, structured file organisation and documentation, version control and make). The outcome was really nice.” (108)
- “Python class goes too deep too fast. Bash classes should have absolute filenames when they jump through files” (82)

Training

- “Emphasize the in class exercises. Wrap up the workshop with automation (show the students how they can link what they learned in 2 days and use it to process hundreds of files all at once)” (68)
- “Maybe more challenges where people try to do what they have just been taught so that learning is reinforced. Especially with Python, getting them to write a simple piece of code to achieve something would help people leave a workshop with a real feeling of accomplishment.” (81)
- “I still want to try an Open Space Technology [1] or other self-organized, multiple-subgroup approach to teaching a workshop. I think that would help motivate the students to take ownership of their learning, and give sub-group leaders an easier time by avoiding extreme breadth in student experience. Of course, you'd need a handfull of knowledgeable leaders available, and a pool of students motivated enough to pro-actively decide what to discuss next. [1]: https://en.wikipedia.org/wiki/Open_Space_Technology” (87)
- “I think shell and git could easily be separate modules. Otherwise people might end up sitting through the same content multiple times.” (Also, shell makes no sense to be taught with MATLAB) The git lesson I think should be taught using a GUI. I've seen too many people go home confused and overwhelmed that have never used it again. It's a bit tragic, because if there is one thing everyone should be using, it's version control.” (89)
- “There needs to be more formal mechanisms for recording lessons learned. There is currently no place for a new instructor to easily learn about what we've tried before and what we want to do in the future. As my main example, many (most?) instructors thought our lessons labeled "novice" were for learners with previous programming experience (i.e. "intermediate"). We need to build institutional memory.” (107)
- “I think it's great that there are well-maintained lesson plans for each of the core topics and for a few other topics (e.g. SQL). However, in the novice workshops I've co-instructed / helped during, it seems that there is **never** enough time to cover the material for a given lesson in a half-day. It would be nice if there were a set of official "abbreviated" lessons so that instructors could have a sense of what could serve as a lesson for a slow / super-novice / installation-issues-ridden class, being able to add in additional content from the "full" lesson repos.” (139)
- “Make the lessons as simple as possible in the first few sections, reusing the same data in different ways. Do not change the dataset until the tool/approach requires it (eg. the SQL lessons uses a more complex dataset than required for simple tasks, confusing the demonstration of the function/tool with learning what a database is). Get each language to solve exactly the same problem, in exactly the same way AND in the 'best language specific way' to show the difference (eg. the R-lesson currently does not use a very 'R' way of programming). Include all the lessons datasets in the SWC virtual machine (so no further downloads are required).” (161)
- Maintain a strong focus on the distinctive pedagogy of SWC and make it part of the core values of the organisation. I specifically mean general philosophy of being friendly to novices, and the specific techniques of sticky notes, MCQ, etc. These are core strengths and I think SWC would benefit from

holding instructors more accountable to upholding them. There is also an urgent need for long-term tracking of instructors and learners to identify exactly how people are benefiting from SWC workshops. This would be a great data science ethnography project for one of the Moore-Sloan centers (UW, UCB, NYU).” (174)

- “More straightforward method of tracking changes to methodology, lessons etc.. I only run courses 3-4 times/year. It is very time consuming and frustrating to have to rediscover the wheel every time I run a course, when everything seems to usually have changed; from how the lessons are structured to how one sets up a SWC website.” (180)
- “Less rote copying/more original thinking in courses (though of course this is hard with large and/or diverse groups).” (181)
- “Flexible/modular examples to customize bootcamps to particular domains” (195)
- “Make the MCQ's easily presentable to students on an overhead projector - currently I screen capture and enlarge!” (198)

Relationship to Universities

- “I think SWC fills a crucial gap in training at the moment. Long term, I think it would be important to transfer much of the responsibility for this training to academic departments that have the resources to offer classes in coding to their students and training workshops to faculty, and who should be preparing their students for the research they are conducting--which now *requires* coding, but departments are failing to provide that training! Coding should be standard for most disciplines, like training in research ethics is now being provided by universities (often by graduate schools to all graduate students, rather than specific departments). In support of this long term outcome, I would like to see SWC promoting these academic courses as a natural extension of SWC workshops. For example, universities can begin with workshops, and use that as a starting point toward developing their own courses. SWC could support current instructors who work in academia to propose courses to their departments, e.g., providing template letters that can be used to propose semester-long courses or seminars, template syllabi that can be adapted to different specific disciplines (and perhaps make a repository of discipline specific syllabi), a template powerpoint presentation with facts/benefits/etc. for drumming up faculty support, and so on.” (22)
- “Work together with scientific disciplines to integrate SWC in their curricula. Liaise with professional bodies to recognise SWC as part of continuing professional development. Train more instructors, run more workshops, gradually develop subject-specific materials/examples/capstone projects.” (30)
- “It would be interesting to see software carpentry work its way into formal classrooms, for university credit.” (149)
- “A bit more clarity and consistency from SWC on procedures and expectations from instructors and partner institutions would improve our ability to collaborate with SWC more effectively, and in a way that would benefit the students in workshops, and the instructors involved.” (169)

Workshops

- “Need more coordination on how workshops are organized. For e.g., there are occasions where targeted audience wasn't clear. This makes it harder for lesson planning. It will be great, if SC can directly involved in getting the reimbursement of the instructors.” (29)
- “Run more workshops so that we have the opportunity to get more and more people teaching.” (33)
- I would like to see more SC workshops in the American Southwest and Mountain West regions.” (80)

Miscellaneous

- “For those of us who are not actively involved in the curriculum development but teach fairly often, it can be challenging to stay up-to-speed on how the lessons and workshop structure are changing, particularly with regards to expectations of minimum topics now.” (21)
- “Nicer landing-page for potential customers... And then use the existing website for instructors and organizers...” (48)

Value of SWC to Instructors Report

- “Instructor training, and building SW/DC international and local communities is important, as it will get people beyond their first (often excellent) student/helper/instructor experience. It will take time for people to either become well versed in the content of teaching wisdom; by enabling continuous support, that time will be shortened.” (98)
- “Stop wasting time on 'red hammer / blue hammer' discussions. These are perennial but add little.” (133)
- “Tips & tricks book for instructors containing all tricks Greg shows during the instructor training and other tricks/tips from blogposts like this: <http://software-carpentry.org/blog/2015/03/whats-in-your-bag.html>” (63)
- “Coffee at the morning sessions?” (182)
- “We cater a lot to data consumers (i.e., researchers that work with existing data), but not so much to data producers (i.e., researchers that produce a lot of data with their programs - for example, simulations of physical systems). This is one area that I'd like to see our lessons explore, and I'm happy to be part of the effort.” (190)
- “Clearer pictures of how to run a SWC workshop if you've never run one. -Tiered levels of instructors (maybe the lead instructor is only drawn from level 2) -Better feedback from workshops to lessons -Better information for instructors about recent developments in SWC - new instructor training techniques, new ways we teach lessons, etc. -Streamlines instructor training for people who trained a long time ago to brush up on their skill/get new -Better assessment of students coding skills in the language being taught” (210)
- “I think we need a more streamlined installation process (I know, there have been lots of progress, but it's still too error-prone - and every minute spent solving installation problem is lost for the whole class)” (214)
- “I think we need resources to continue further learning after workshops, facilities to build physical and virtual communities where software carpentry participants can share ideas, expertise and initiatives. For example, prepare and publish guidelines for creating and maintaining communities around promoted skills. Also something like the "Software Carpentry Day" in which could be carried out promotional activities simultaneously in different parts of the planet should be interesting.” (113)

Appendix – Methods

There were a total of 215 responses. Respondents were recruited via two emails sent from the Executive Director to the entire pool of SWC/DC instructors. Respondents were asked to respond online to an open-ended survey and/or include responses via email if they preferred.

To analyze the data, the researcher coded the responses inductively, meaning that she grouped responses that were similar. One way to understand this process is to imagine you were dropped into a new ecological environment and were told to create a taxonomy for all the alien plants you see. You have a general idea of how to do that (trees vs. grasses) but the specifics of these plants are unique. The researcher develops codes based on what they see in the plants in front of them, and then applies those codes to each new specimen. When groupings emerge, new taxonomic codes are created and the researcher goes back to recheck older coding. The process is iterative and relies on previous data and understanding.

There are a few things to remember when reading the results of qualitative data of this kind.

- **Codes may include other codes**
One person could respond generally and another specifically. For instance, one might say “I learned how to code” and another “I learned how to write scripts in Python”. While they may essentially be the same thing, because we don’t know if learning how to code was in Python, they are coded separately.
- **Codes are not mutually exclusive**
Statements often include more than one idea or response to the question. Therefore the majority of responses in this dataset led to more than two codes and as you can see in the responses, the quoted responses are often indicative of more than one code.
- **These codes are not perfect**
With more resources, multiple people could code these and then we would test inter-coder reliability to hash out the best and most descriptive coding scheme building on previous work. This is a pilot study with one researcher doing the coding and as such it’s imperfect.
- **Codes may not indicate all that a respondent agrees with**
What one respondent says may not indicate all that they believe to be true about the answer. For instance, a respondent might have answered a question differently if they had been given a list of possible answers. Simply not having a code doesn’t mean that the respondent disagrees with the other answers.

For those reasons, frequency counts in qualitative coding can be misleading and the “one off” codes (meaning only one or two people said them) can offer useful information. To illustrate the codes and to help make sense of the responses for questions 1 through 4, I have selected illustrative or well-worded quotes. These give an idea of what others said. The statements can (and do) span several codes even if they are

listed under one code for illustration.

All responses have been included in question five, which asks for feedback. As there were less clear patterns, I grouped the feedback into categories and did not summarize the sections.

All quotes are reproduced as written and without editing of the content. Some editing has been done to reduce long paragraphs or divide comments that fall into several categories but care was taken to not edit the language or words used. In qualitative research, best practices are to preserve the writing style as it is, regardless of typos, spelling mistakes or grammatically incorrect statements.

The public version of this report omits anyone who responded “No” to the question “Do you give your permission for your response to be used in promotional materials?” or who requested to be informed if their information was used.