
Welcome to The Carpentries Instructor Training!

Sign in: Name (Pronouns), Institution, Email & Socials (optional)

Please sign in so we can record your attendance.

- Jan-Hendrik Müller, he/him, Universität Göttingen, jan-hendrik.mueller@gmx.net, https://github.com/kolibril13
- Madison Golden, University of Utah, madison.golden@utah.edu
- Laura Schild (she/her), Alfred-Wegener-Institut Potsdam, schildlaura@outlook.de
- Benjamin Keene, UCF, keene@ucf.edu
- Sonja Grath (she/her), LMU Munich (Germany), grath@bio.lmu.de
- Florian Kohrt (he/him), Ludwig-Maximilians-Universität München, fkohrt@anche.no
- Carleigh Young (she/her), University of Wisconsin-Madison/Fairmont State University, cbyoung6@wisc.edu
- Morris Greenberg (he/him), University of Toronto
- A.J. Stein, NIST, alexander.stein@nist.gov, https://github.com/aj-stein-nist/
- Cory Thomas, Aberystwyth University
- Elviss Dvinskis (he/him), TU Delft, e.dvinskis@tudelft.nl
- Nicholas Owen, UCL, n.owen@ucl.ac.uk, @Dr__NO, https://nicholas-owen.github.io/
- Dimakatso Maheso (she/her), University of Johannesburg, d.j.maheso@gmail.com
- Emily Zhu (she/her), Texas State University, c_z88@txstate.edu
- Angelia Miller, she/her, University of Massachusetts Dartmouth, amiller7@umassd.edu, https://angeliamiller.github.io/
- Gokmen Kilic, he/him, University of Durham, gokmen.kilic@durham.ac.uk
- Elissah Becknell, she/her, Minitex/University of Minnesota, ejbeck@umn.edu
- Rebekah Silverstein (she/they), rebekah.silverstein@okstate.edu
- Florencia Mangini (she/her), TeamUp
- Amber Chen (she/her), amber.chen@psych.ucsb.edu
- Norman Hebler (he/him) nhebler@sun.ac.za
-
-
-
-
-
-

-
-

If you have a moment before we begin and have not yet done so, please fill out the pre-training survey at https://carpentries.typeform.com/to/QVOarK#slug=2024-03-26-ttt-online-CET

You can keep track of the time in your current timezone at https://timeanddate.com/worldclock.

Break times (approximate): 14:30 CET (13:30 UTC)

Welcome

https://carpentries.github.io/instructor-training/01-welcome
Questions:

- What is The Carpentries and how do we approach teaching?
- What should you expect from this workshop?

Objectives:

- Identify common ground with some of your fellow workshop participants.
- Understand a general structure and core goals of The Carpentries.
- Predict what will and will not be covered in this workshop.
- Know where to find The Carpentries Code of Conduct and how to report an incident.

Code of Conduct:

To make clear what is expected, everyone participating in The Carpentries activities is required to abide by our Code of Conduct.
https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html
Any form of behaviour to exclude, intimidate, or cause discomfort is a violation of the Code of Conduct. In order to foster a positive and professional learning environment we encourage you to:

- Use welcoming and inclusive language
- Be respectful of different viewpoints and experiences
- Gracefully accept constructive criticism
- Focus on what is best for the community
- Show courtesy and respect towards other community members

If you believe someone is violating the Code of Conduct, we ask that you report it to The Carpentries Code of Conduct Committee by completing this form: https://goo.gl/forms/KoUfO53Za3apOuOK2

Introductions

Anne Fouilloux:
Rob Davey:

Exercise: Getting to know each other

- If the Trainer has chosen an icebreaker question, participate by writing your answers in the Etherpad.

- How do you hope becoming a Carpentries Instructor will change your approach to research and/or teaching?

- - Rob/Anne
- - I would like to be more effective and efficient in my teaching and eventually have (again) more fun while doing it. I am a Bioinformatician by training (BSc/MSc), have a PhD in Molecular and Evolutionary Biology and have been teaching a wide variety of courses at University level for more than 10 years (lab course for bioinformaticians at one end, R for Biologists at the other end, for example).
- - I'm actually just getting started as a TA and am wanting to be able to adequately teach my students especially with topics like data/R/python as they feel they aren't getting the teaching they need
- - I am hoping to be able to teach with more hands-on activities and move away from 1-shot lecture-style classes. I also hope to make programming more accessible/ less intimidating
- - I would like to learn how to make the coding skills and computational techniques more available in psychology.
- - I'm planning some YouTube Tutorials about OpenSource Software packages, and I hope to use some Carpentries approaches there.
- - I would like to make sure as I am growing in my field (current PhD student in statistics) on the theoretical side that I can still be accessible in teaching and not too in the weeds for introductory and intermediate coursework, while still introducing good habits and proper rigor
- - I'd like to improve the way I teach and understand better which methods work. My goal is to give studnets and peers the skills they need to be better scientists. Additionally I hope that becoming a better teacher will also help with becoming a better communicator and collaborator to improve my own scientific work.
- I've been an instructor at a community college for 14 years (Library Information Technology Program) and recently switched jobs to become a trainer for library workers (Cataloging & Metadata skills, technologies, and workflow). I worked with many young adults at the community college, but now I'm working more with middle aged and older adult students. This seemed like a good training for working with highly motivated library workers who want to up their tech skills.
- I hope to get more insight on various teaching techniques and how to easily adapt to new teaching methods based on my audience/class
- I am hoping to feel more comfortable at the classroom environment while teaching and also feel to learners same way. I am currently swiched from industry to academia so that i would be greart to know about how to manage learning and teaching skills.
- I (Florian) hope to learn about "low hanging fruits" that can improve my teaching, like what to always include and what to better avoid. Also, I hope what I learn generalizes to other settings, e.g. giving good feedback or structuring my own learning paths
- I have a background in molecular biology and am not very familiar with teaching, so all of this is new to me. My only real teaching experience comes from essentially teaching lab work, which I believe is completely different. Therefore, I consider myself almost a blank slate. So I guess what

I learn here becomes my way of approaching teaching.

- I hope to expose analytics to a broader audience than college classroom with the help of Carpentry training, so that more people can benefit from the emerging analytics techniques.
- I hope to learn more about how to teach to others. In my current position, I need to work with recent Graduates developing their first software projects. It seems a good training to understand how to help others.
- I hope becoming an Carpentries Instructor will improve the way in which I talk about data, my research, and my workflows more succintly as well as how to encourage open data science mentality at my university or in future positions
- - I have a lot of college-level mathematics instruction experience, I'd like to develop my programming/linux/data science teaching skills. Members of my GRA team are Carpentries instructors, and they love it, so I'm looking forward to learning what I can.
- - So I can offer software carpentries workshops within my university, as there currently lacks any courses for research students outside typical lectures.

I recently changed roles to a Data Steward after many years as a postdoc fellow, training for research is not always available to new researchers, many reinvent the wheel, however I am in a position now to enhance researchers experiences through training them in best practices etc. I have a focus on biosciences and bioinformatics so the Carpentries was a logical step to having a recognised standard to train others with.I also wish to develop new courses on specialised topics researchers need.
I feel like teaching for the carpentries will be a perfect way to explore teaching at university. I am a new faculty member at OSU Stillwater, and the carpentries are used regularly in the library. I enjoyed the lessons I participated in while completing my MLIS, and I hope to share these experiences with students and my colleagues.
- I teach people new technology and tools to outsiders and I am usually the "git wizard" for insiders. I want my teaching of technology to be outcomes-focused and more productive like when I did TEFL training years ago.

Workshop schedule and break times

-- -- *Exercise: Reviewing The Carpentries Experience and Goals* -- --
For the multiple choice questions below, please place an “X” next to the response(s) that best apply to you. Then find yourself a spot in the Etherpad below to write a short response to the last question.

1) Have you ever participated in a Software Carpentry, Data Carpentry, or Library Carpentry Workshop?

- Yes, I have taken a workshop. xxxxxx
- Yes, I have been a workshop helper.x xxxx
- Yes, I organized a workshop.x
- No, but I am familiar with what is taught at a workshop. xxxxx
- No, and I am not familiar with what is taught at a workshop. xxxxxxxx

2) Which of these most accurately describes your teaching experience?

- I have been a graduate or undergraduate teaching assistant for a university/college course.xxxxxxxxx

- I have not had any teaching experience in the past.x
- I have taught a seminar, workshop, or other short or informal course. xxxxxxxx
- I have been the instructor-of-record for my own university/college course. xxxx
- I have taught at the primary or secondary education level.xxxx
- I have taught informally through outreach programs, hackathons, libraries, laboratory demonstrations, and similar activities. xxxxxxx

3) Why are you taking this course? What goals do you have for today and tomorrow?
I want to organize my teaching in a better and more efficient way and eventually teach R for biologists at our university better as it is taught at the moment.
I am taking this course as a part of a grant-funded project to incorporate open data science practices/learning/collaborations within the University of Massachusetts learning network and to better improve how I discuss the data life cycles, and technology. My goal is to learn a skill on how to motivate learners.
I volunteered with colleagues (also instructors) to run Carpentries Workshops. (And I wanted a better approach to teaching git and commonplace software tools.)
To be able to host workshops for for the organisation I am working with
I want to learn more about the science of teaching.
I am taking this course because of the deparmantel teaching requierments that they want me to teach sometime. My goal for today being familar with all concept and having a demo teaching session for tommorrow.
I am working towards being an instructor so I can teach workshops at my library/ through my institution. I look forward to learning more about carpentries teaching methods.
I have prepared courses and held trainings in IT related matters before (reproducible data analysis, R programming, Quarto, basic IT security), but so far have not had any pedagogic training in this regard. Although I have picked up a few best practices (such as “never do stuff on behalf of the participants”, or “let them work in teams”), I would love to become better equipped in order to improve my teaching. I would like to learn about "low hanging fruits" that I can easily implement in my teaching or about things I could easily avoid.
I would like to learn how to make the coding skills and computational techniques more available in psychology. I find it difficult to make my workshops suitable for coding novice and would like to learn how to take the perspectives of the learners.
I want to bring Carpentries to my students and to be a better technology teacher
To participate/lead/develop Carpentries courses at my uni, as well as focus on how to maximise teaching success and not have to repeat everything every time new ppl start. Need to develop materials in the correct manner.
Improve my skills in structuring, presenting, and explaining data and software to people. My goal is to convey information in a clear and engaging manner, making it accessible to a wide audience, regardless of their technical background. I believe that enhancing these skills will enable me to communicate more effectively.
I would like to collaborate with other instructors, and host a Carpentries workshop that could benefit students at my school (and from elsewhere). I would like to learn new teaching strategies, and become more comfortable teaching programming/linux concepts via Zoom. My goal for the next few days is to soak up all that I can, and immerse myself in the Carpentries community!
To become more familiar with the Carpentries pedagogy, and become more comfortable teaching my colleagues. I would also like to help develop a data wrangling and data viz for librarians learning module.
I would like to be able to assist in carpentry training sessions happening on campus and provide future training.

I'm interested in technology skills builidng approaches, especially in workshop style courses (as opposed to formal classroom instruction)
I'm attending this training as a recommendation by the data science academy of the Helmholtz Association with the goal to provide courses more frequently and teach data and software science courses for other scientists.
I'm attending this workshop to develop my teaching skills but also helping my students. Also I heard excellent suggestion from my collegues to take this workshop.
To actually recieve somesort of formal teaching training.
To refresh on best practice of teaching and potentially get something new. Be well equipped to organize workshops.

This exercise should take about 5 minutes for responses, with an optional 10 for additional discussion as time permits.

A Brief Overview of the Carpentries

Image: Action figures in a workshop with Instructor, Co-Instructor, Helper, and Sticky Notes labeled https://carpentries.github.io/instructor-training/fig/Scene_1_blue_stickies_labeled.jpeg

- Software Carpentry focuses on helping researchers develop foundational computational skills
- Data Carpentry focuses on helping researchers work effectively with their data through its lifecycle
- Library Carpentry focuses on teaching data skills to people working in library- and information-related roles.

Instructor Training Overview

- How learning works
- Building teaching skill
- Creating a positive learning environment
- Carpentry history and culture

What We Leave Out

- workshop content & technical skills
- how to develop new lessons

What Questions Do You Have?

- Is it only me who feels completely uncomfortable teaching exposed myself in front of a screen with almost all black screens? I actually want my students to be physically back IN the class room ;-)

Keypoints:

- The Carpentries is a community of practice. We strive to provide a welcoming environment for all learners and take our Code of Conduct seriously.
- This episode sets the stage for the entire workshop. The introductions and exercises help everyone begin to develop a relationship and trust.
- This workshop will cover evidence-based teaching practices and how they apply specifically to The Carpentries.
- Learner motivation and prior knowledge vary widely, and can be quickly assessed with a multiple choice question.

Building Skill With Practice

https://carpentries.github.io/instructor-training/02-practice-learning
Questions:

- How do people learn?
- Who is a typical Carpentries learner?
- How can we help novices become competent practitioners?

Objectives:

- Compare and contrast the three stages of skill acquisition.
- Identify a mental model and an analogy that can help to explain it.
- Apply a concept map to explore a simple mental model.
- Understand the limitations of knowledge in the absence of a functional mental model.

The Carpentries Pedagogical Model

Acquisition of Skill

https://carpentries.github.io/instructor-training/fig/skill-level.svg *Image: Three people, labeled from left to right as "Novice", "Competent Practitioner", and "Expert". Underneath, an arrow labelled "Experience level" points from left to right. The "Novice" is quoted, "I am not sure what questions to ask." The Competent Practitioner is quoted, "I am pretty confident, but I still look stuff up a lot!" The Expert is quoted "I have been doing this on a daily basis for years!"*

- Novice
- Competent practitioner
- Expert

Mental Models

"All models are wrong, but some are useful."

The power (and limitations) of analogies

-- -- *Exercise: Analogy Brainstorm* -- --
1) Think of an analogy to explore. Perhaps you have a favorite that relates to your area of professional interest, or a hobby. If you prefer to work with an example, consider this common analogy from education: “teaching is like gardening.” Life is like a box of chocolates!

You may also like to think of analogies summarised as "x is to y as a is to b", e.g. Fish : River :: Bird : ____

Put together full sentences to describe the relationship.
"Learning is a marathon and not a sprint." Part of it is right as it takes time to learn a subject properly. However, actually learning is never really over and one could always learn more in any particular field. Probably the point sums up to 'when is it well enough'.
Learning is a social construct. defined: "A social construct is a concept that exists not in objective reality, but as a result of human interaction." This idea is useful, because it allows everyone in the classroom to understand that learning is an agreement, we are agreeing to be together and engage in practices of teaching and learning. I like emphasize the social apect of learning, that we are in relationship as we learn together.
Data are like puzzle pieces, when connected they create information
Learning is try to solve a puzzle. You start to learn how small pieces work together until the moment that you set all together. =)
Knowlege is like a town sourrounded by fog. In the town you're in the confort zone. But as soon as you go out into the fog, there might be new things to discover.
Leanring is like hiking: it can be more flat or more steep and it can also have many routes to the same destination.
Teaching requires those who are ready to absorb knowledge like gardening requires to absorb water, light and nutrients
Infinite like a universe you only can see the sky.
An IDE is to coding as goggles are to swimmers. You could do without them, but you don't really want to.
When you teach others you also learn with others.
Learning to code is like going through a labyrinth. When you're learning to code it feels like you're constantly running into the same walls again and again but eventually you get through it and when you go back into the labyrinth you get better oriented each time you go back in and come out (fortunately this labyrinth doesn't have traps or portkeys)
Life is like riding a bicycle. To keep your balance, you must keep moving forward.
Don't be afraid to click on all the buttons you can't break the internet. like a child on a playground trying out all the swings, slides, and seesaws. Be curious about the new tool you are trying, be playful and exploratory, and have fun, lean into the freedom and joy in playfully learning and discovering new concepts.
learning a task is like a maze, although there may be many paths the objective is the same
Databases are like your Spotify playlist. You should care about the design and structure as much as you do about your playlist.
Data visualizations are like jokes. If you have to explain them, they probably aren't very good. Not

entirely true, there exist complex visualizations that require some additional explanation -- however, it's a fair rule of thumb for communicating information visually.
"A strong password for your account is like a strong lock and key on your home's door." In cyber-security, there are many popular analogies, and this one comes up often. Like all good security analogies, we often focus on what happens when the analogy's conditions is false, not when it's true: using a poor password or no password is like not locking your door. We are supposed to imply that, from personal experience, we should know it is good to lock the door with a key when we are not present. The password is our key to the metaphorical door of an account with our sensitive data. We don't want a key that it is too simple to create or recreate without a lot of effort, and hope a person trying to get into the house will give and attempt unlawful entry into another house.
encryption is like putting something in a vault (came up with this before hearing A.J.'s thoughts on this, which I do find valueable)

- you wouldn't leave the key in front of the vault (don't put your password on a sticky note next to cour computer)
- you wouldn't use a vault with a simple key for a warded lock (make your password long)
- you wouldn't trust a brittle door (use a trusted software/algorithm)
- you wouldn't leave the door to the vault open (shut down the computer to enable full-disk encryption)
- you wouldn't leave copies outside the vault (have every copy encrypted)
- if you had a vault, you wouldn't hide your jewels instead (use encryption rather than obfuscation)

2) Share your analogy with a partner or group. (If you have not yet done so, be sure to take a moment to introduce yourself, first!) What does your analogy convey about the topic? How is it useful? In what ways is it wrong?

Analogy Examples:
-

This activity should take about 10 minutes.

Analogies at Work: “Software Carpentry”

https://carpentries.github.io/instructor-training/fig/mental_models.svg *Image: Three collections of six circles. The first collection is labelled "Novice" and has only two arrows connecting some of the circles. The second collection, labelled "Competent Practitioner" has six connecting arrows. The third collection, labelled "Expert", is densly connected, with eight connecting arrows.*

Concept maps
A mental model that a young child might develop after placing a ball in water:
Image: Two words inside rectangles, with labeled arrows connecting them. "Ball" is at the left, with an arrow pointing to "Water", at right, labeled as "Pushes out." https://carpentries.github.io/instructor-training/fig/ballwater1a.svg

A more complex mental model the child might construct after playing with balls of different sizes:
Image: Four words inside rectangles, with labeled arrows connecting them. "Ball" is at the left, and "Water", at right. "Big Ball" and "Small Ball" are stacked vertically between them. Arrows from "Ball" are labeled "can be MORE" and can be "LESS", and arrows to "water" are labeled as "Pushes out MORE" and "Pushes out "LESS" https://carpentries.github.io/instructor-training/fig/ballwater2a.svg

-- -- *Exercise: Mapping a Mental Model* -- --
1) On a piece of paper, draw a simplified concept map of the same concept you discussed in the last activity, but this time without the analogy. What are 3-4 core concepts involved? How are those concepts related? (Note: if you would like to try out an online tool for this exercise, visit https://excalidraw.com .)
2) In the Etherpad, write some notes on this process. Was it difficult? Do you think it would be a useful exercise prior to teaching about your topic? What challenges might a novice face in creating a concept map of this kind?
This exercise should take about 5 minutes.

Misconceptions

Our child might assume that ball weight and ball size can both be described by the mental model developed by playing with balls of different sizes.
Image: A concept map similar to the previous one except with "Heavy Ball" and "Light Ball" in the middle, and a red "X" over the arrows labeled "Pushes out MORE" and "Pushes out LESS"
https://carpentries.github.io/instructor-training/fig/ballwater3a.svg

It may take a while to adjust, but eventually new understanding will coalesce:
Image: A new concept map. "Ball" remains at left, and "Water", at right. "Size" and "Weight" are stacked vertically between them. Arrows from "Ball" share the label "Can have more or less." One arrow from "size to "water" is labeled "Affects pushing of"
https://carpentries.github.io/instructor-training/fig/ballwater4a.svg

The process of forcing abstract knowledge into a visual format can often reveal connections you may not have been aware of, or illuminate gaps. This can be especially useful when preparing to convey aspects of your mental model to someone else!

Misconceptions:

- Factual errors
- Broken models
- Fundamental beliefs

-- -- *Exercise: Anticipating Misconceptions* -- --
Describe a misconception you have encountered as a teacher or as a learner.
This exercise should take about 5 minutes.

As teacher: "All learners in a classroom/workshop are actually there to learn about a subject, gain knowledge and are interested in the overall topic." - not necessarily true, in particular for mandatory classes (the participants might actually only be there as they have to and they are not interested in the subject at all and/or only want to get a good grade with the least amount of work possible

As a teacher (Mathematics): Students often struggle to graps how interconnected theorems/definitions/concepts are. Their math 'workflow' looks like a directed flowchart (step 1 -> step 2 -> ...), and have difficulty scaffolding their preexisting knowledge to what they are learning now. Their misconception includes a partly broken model about the interconnectivity of math, viewing individual problems as wholly independent from everything else. Analogous to learning only common tourist phrases "Where is the ... ", "My name is ... ", etc., when visiting a foreign country and attempting the language. A helpful technique for breaking this is finding a correct "ground truth" the student has and asking them Socratic/probing questions to lead them in the right direction. But in general this is a widespread problem as a teacher, and it exists in myself as a learner when working in a field I'm unfamiliar with.

As a learner, motivation can be a tough thing. Sometimes we are faced with stuff we must learn and not stuff we want to learn. Having an engaging instructor and compelling/organized curriculum is essential for unmotivated learners. Also, have clear learning outcomes- for those students who are working towards to final result and not treating learning like a enjoyable journey.

As a learner, general concepts and applications of code to example data is good to show how the code works and what it outputs, but the practice and utility does not always stick or easily recalled until you need it in your own work. This could be said a lot for general theories and concepts as well in that what is good or works for the majority, might not always apply for the one.

I create, maintain, and use information modeling tools at work. The tool encodes concepts, so there have been many misconceptions with coworkers and community members at work. Most of them follow a general pattern: one person or group has a very precise definition of a concept and how it fits into the model, and others want a more flexible definition. Most of the time, both parties believe there is an objective truth for their position and itemize anecdotes and generalize from this anecdata.

As a teacher of informal workshops, I've had the misconception that just because someone has a technical background does not mean they will master a technical topic quicker.
As a learner, I've often had the misconception that something is much more complicated than it is. For example: learning to knit was very daunting to me and I thought it would be really hard, but its really just one movemement over and over again.

As a teacher, "one size suits all", with regards to how material is taugh, in reality people learn in different ways.

In my analytics course, students generally think "big data" refers to datasets of a large volume only, while overlooking other characteristics of "big data", such as volocity, complex formats ect.

As a learner, I've noticed that sometimes there can be misconceptions on the teacher's side regarding the varying levels of ability within the classroom. This often results in the teaching level being adjusted to accommodate those who are more vocal and engaged in discussions with the teacher. This adjustment can happen in both directions, either leveling up or down. This situation underscores the importance of having materials prepared beforehand, conducting pre and post surveys, etc.

As a learner, I feel like asking questions might reveal me as a novice when the opposite is true. When you ask questions it brings light on your misunderstanding sure, but it also allows for others in the room to learn and gain clarity and helps to open up the conversation. Questions can be great teaching tools.

Using Formative Assessment to Identify Misconceptions

How can you prevent hidden misconceptions from interfering with learning? Seek them out with assessment!

Formative vs Summative assessment

-- -- *Exercise: Formative Assessments* -- --
Any instructional tool that generates feedback that is used in a formative way can be described as “formative assessment.” Based on your previous educational experience (or even this training so far!) what types of formative assessments do you know about?
Write your answers in the Etherpad; or go around and have each person in the group name one.
This exercise should take about 5 minutes.

This exercise should take about 5 minutes.

-- -- *Exercise: Identify the Misconceptions* -- --
Choose one of the wrong answers to the question below and write in the Etherpad what the <u>misconception</u> is associated with that wrong answer.
Q: what is 27 + 15 ?
a) 42
b) 32
c) 312
d) 33
This discussion should take about 5 minutes.
(d) 312 = 5+7 (12) 1+2 (3) -> 312, the basic misconception is perhaps not putting the correct digit under each other (not sure how to explain this properly, was one of the standard mistakes in primary school)
(b) not taking over the '1' when adding up 5+7 (12) -> leads to a 3 instead of a 4
(b) 32 = 2+1=3 and 7-5=2, the fundamental misconception is that we are treating 27 and 15 as independent groups of tens and ones (not sure how to explain that any more succintly). Then we are adding tens and subtracting ones to arrive at this answer, but we should treat these as whole numbers to add together.
(c) 20 + 10 = 30; 7+5 = 12; 30 and 12 are concatenated to 312 instead of summed
b. could be the answer if subtraction and addition were used interchangeably. if you subtract 5-7=2 and 2+1=3 giving 32.
d) 2 + 1 = 3 and 7+5 =12 then sum up the all 15 could cause the nearst answer is 33
(d) taking the 1 (from 5+7 = 12) over but actually putting it on the wrong digit
(b) Summing 2 and 1 and 7 and 5, but not carrying over the 1 from 12.
b) Adding 2 + 1 = 3 and then adding 7 + 5 = 12 but not carrying over the 10, person thinks the quick anser is 32. Hope it makes sense.
c) 7+5=12, then 10+20=30. They might combine both parts together being 312.

(C) 7+5=12 anbd 2+1=3 > putting these together creates 312. You can tell the student was adding up the numbers in "chunks" and didn't add the two outcomes together properly.

(d) 27 + 15 = 33, student likely struggles with place values in addition. Specifically, when adding the ones digits, 7 + 5 = 12, the student might place the tens digit '1' on top of the ones place in the sum, so instead of representing 12 it represents 1 + 2.

(b) 32 = 27+5 , missing the last part where the 10 has to be added as well.
c) 312, 2 + 1 = 3 and 7 + 5 = 12 hence the answer is 312
Formative assessments are most powerful when:
1) all learners are effectively assessed (not only the most vocal ones!) AND
2) an instructor responds promptly to the results of the assessment

-- -- *Exercise: Handling Outcomes* -- --
Formative assessments allow us as instructors to adapt our instruction to our audience. What options do we have if a majority of the class chooses:

- mostly one of the wrong answers?
- mostly the right answer?
- an even spread among options?

Choose one of the above scenarios and compose a suggested response to it in the Etherpad.

This discussion should take about 5 minutes.

The Importance of Going Slowly
"If someone feels it is too slow, they will be a bit bored. If they feel it is too fast, they will never come back to programming."

Meeting learners where they are

Keypoints:

- Our goal when teaching novices is to help them construct useful mental models.
- Exploring our own mental models can help us prepare to convey them.
- Constructing a useful mental model requires practice and corrective feedback.
- Formative assessments provide practice for learners and feedback to learners and instructors.

BREAK (15 min)

Expertise and Instruction

https://carpentries.github.io/instructor-training/04-expertise

Examining Your Expertise

You may not think of yourself as an "expert" but many advantages -- and pitfalls -- may apply to you.

What Makes an Expert?

-- -- *Exercise: What Is An Expert?* -- --

What is something that you are an expert in? How does your experience when you are acting as an expert differ from when you are not an expert?

- Spent at least 1000 hours on it and feel to teach the topic any of people the area e.g kids, highschool level, etc.
- I believe I'm an expert at hobby electronics / computing. The key difference between this area and others is when something breaks and I have to fix it. If it's electronics related, I have a decent idea of how to approach repair. Other fields I run crying to another expert. The biggest difference between expertise/lack of expertise in my experience is having a good idea of "where to begin".
- An expect has spent a lot of time on a particular thing
- I'm expert in cooking, I know several recipes by heart, can improvise with ingrediants in my fridge, practice on a daily basis, and have basic techniques for preparring food down (chopping, saute, etc.). I find it difficult to teach cooking to my husband because he lacks some basic skills and gets frustrated with the time it takes to prepare something without fast knife skills or ability to swap ingrediants (because you know that cornstarch works the same as arrowroot powder). As an expert in cooking with little formal training, I lack a curriculum or organized way to approach teaching my husband something that is complex.
- Experts have lots of experiences and can handle situations when things change and provide instructions for new people.
- I'm an expert at cozy style games and now that my husband is starting to get into cozy games it's interesting to see how much difference there is even for something that's a casual level type hobby - but I can tell there are areas that I consider 'common sense' or 'that's usually the case' when he may not know that. On a professional note I'm the tech expert in my office with CRM/data stuff but compared to IR or App Services I'm not the expert
- Expert in human rare disease research, 30+ yrs, 10 years bioinformatics, expert in omics, RNA splicing, R. Experience, understanding of the subject field and areas related, ability to adapt using prior knowledge.Can lead through knowledge.
- Expert in Open Source Development. Novice might now yet know: What are common workflows, what are the communication channels, what to expect from software and how to evaluate ideas
- Research data management. I am able to answer any questions on the topic regardless of how in-depth or discipline-specific they seem. When I am answering questions or trying to problem solve in areas that I do not have expertise, it requires more research and inquiry.
- Transposing or "copy-paste-modify"ing my mental model from a related field (earlier material was spot on and resonated with me; I am guilty of that all the time!), so I often ask about analogies and relate terms/concepts from other specialties in cyber-security and software development I am "expert" in.
- I struggle a bit in defining what 'an expert' is as to me it only works in comparison to other people. For example, I am probably an expert in cooking compared to my 10y-old son, but not compared to Jamie Oliver. Taking this example (cooking): when I am about to teach my son how to make pancakes I learned that I have to define some things first that are by now completely obvious to me, for example, that the exact bowl to make the dough in does not really matter
- Being an expert does not necessarily means to be a good teacher - we probably all had bad math teachers who were succeeding in their subjects but could not understand that other people might be even having trouble with it
- I consider my self an expert in the makerspace field but I'm always learning form others and this is key to staying relevant in a field of expertise.
- an expert knows how to deal with confusing situations, i.e. how to systematically come to an understanding
- I am pretty good knitter and can get into new projects quite easily. However, trying to teach someone else is pretty hard for me as manual skills are difficult to put into words so I end up

having people just watch me.

- I ran a workshop a few weeks ago, teaching both introduction to python and R. You could easily see the difference between the lessons, where python (expert) was much more fluid and could anwser questions and proivide additional insight/ or way around the problem. Whereas for R, I could just about anwser the questions but it was minimal and the lessions were far less fluid. this workshop was a nightmare due to the range of abilities.
- Would consider myself experienced in structural biology, there are not a lot of people in the field and because of that I get some sort of reassurance that I am capable and knowledgeable in that particular niche. But in the broader field of molecular biology I would rather listen and observe, and continue learning.
- As an expert, you are more readily able and willing to jump into conversations/discussions regarding a specific subject versus as a novice you're more interested in the act of doing or listening in particular situations. For example, as a long time soccer player (would not consider myself an expert because I don't play pro), I am more readily able to direct teammates and game play, and explain concepts or other strategies teams are employing to help our team advance and defence successfully on the pitch.
- Experts understand where to find the information and concepts and can see value in the practice and are interested in expanding their field of expertise.
- I become in a expert in different ways to teach languages, so I feel comfortable doing that. On the other hand, it is notably when I am not familiar with something.

This discussion should take about 5 minutes.

Experts have more connections among pieces of knowledge.

Image: Three collections of six circles. The first collection is labelled "Novice" and has only two arrows connecting some of the circles. The second collection, labelled "Competent Practitioner" has six connecting arrows. The third collection, labelled "Expert", is densly connected, with eight connecting arrows. https://carpentries.github.io/instructor-training/fig/mental_models.svg

Expertise and Teaching

Mind The Gap

-- -- *Exercise: Awareness Gaps* -- --

- Is there anything you are learning how to do right now? Can you identify something that you still need to think about, but your teacher can do without thinking about it?
- Think about the area of expertise you identified for yourself earlier. What could a potential awareness gap be?

Learning to use Kumu for social network analyses that our Professor worked with using a set dataset but trying to apply that to the dataset I'm actually using and understanding the differences in elements vs connections specifically for the software we're using
My awareness gap when reach research data management is the underlying principles and knowledge necessary to understand the best practices we are trying to each. For example, telling someone how often they should perform a back-up necessitates that they already know what a back-up is and how to do it. As a learner, I am working on knitting and crochet. I often find that in tutorials, people assume you have

watched their videos in order and already know things like the names of different stitches, needle sizes, etc. and will do "call backs" like "you should remember from my last video..."
I am actively learning numerical methods for approximation theory in mathematics. My research advisor has the ability to identify upper bounds for many quantities and objects in our research, pulling from various numerical methods. I usually need to rewrite the quantity in a form more similar to the textbook we're referencing, and I may or may not make the connection. Then, I take it to him and he points out the upper bound almost immediately.
I am learning how to apply geostatistical models in my Master's degree research and specifically learning about what quantities and values I can use to choose an optimal model, which my teacher is well versed in and ready at the helm to suggest a variety of processes and how to apply them. An awareness gap from my previous example (an is also applicable here) is that I assume people on my team have the same knowledge of terminology and what they mean and how to carry out the steps that need to be taken based on the terminology

- I'm learning how to do weight training (lifting weights) and it's tough for me to think about how much weight to start a set with. Example: I'm at the Legpress machine and can't just get started. I look at my previous workout notes and start with a ballpark number, then adjust. It's also slow if I sit down and map the exercises and starting weights before I get to the gym. I'm constantly adjusting before I get into a rhythm. It seems to take forever to get through a workout by myself, but with a personal trainer it's much faster!

Constantly learning, that is part of the job and that is great. However, some learning habits are hard to kick, meaning I am sure there are more effective ways on how to approach learning itself. I can only assume that with experience some things come more naturally.
I am learning a new job, how to navigate a new campus, and where to find the information I need to do my job well. I lean on my coworkers who have been at the Library for a long time and try to ask questions daily. I have been tasked with discovering inefficiencies so I have to pay attention and learn about all the data tools that could benefit the team.
Choosing a research topic, my supervisor has been in the field for decades that he can easily see a gap in the Astrophysics research while I kinda struggle a bit to choose a reserach topic based on previous resercah
I'm learning how to make scientific visualizations, and often have to look up object modifiers. A teacher whould know these modifiers already by heart.
I am working in a standards group making a new binary encoding (well, new to me). Others can read the schema models immediately and "grok" and it still takes me reading documentation and 1 or more hours to understand.
I am learning ballroom dancing right now. For our teachers it is easy to think of posture, feet and expression at the same time, but we as learners are still struggeling with each component individually. Shifting your weight, for example, is something that our teachers do automatically, but has to be told to us learners explicitely.
Learning to use visulization and VR for certain academic requests.
I am learning MPI programming and with hands on session it is going well though when it comes the my large codes i have difficulties to implment it mine. So with hands on lessons teacher/instructor can do all stuffs and codes easily but in my case it is not working same way. As my previous role i was assuming that all technical people know the basic managment tools about network as an awareness gap.
Locating relvant structures to describe a breast lesion in a mammogram, radiologist can do it instantly and all I see a bunch of wavy grey lines. the awareness gap, is they understand the underlining structures of the breast and how they translate to the imaging technique. So they can simple look at where they would expect to find said information.
i know i am always learning, life isnt something you can learn it all, so simply put the gaps are always

there to be filled. My current gaps are how to adapt and change from an academic publish or perish mentality to get everything done yday to a more sustainable method of working to support researchers.
I am in a continue cycle of learning, I am learning how to use Python, how to write essays, how to incorporate more tools to my toolbox.
I'm learning to do exercises and loose weight. Those workouts are very easy for my trainer, but I can do them wrong and hurt myself.
Maybe making breaking the thing a challenge, who can break it the best?!

This exercise should take about 5 minutes.

Switching Language

-- -- *Exercise: What do you use interchangeably?* -- --
In the Etherpad, share an example of words or notation that you sometimes use to accomplish or refer to the same thing. If possible, try to think of an example that might occur in a Carpentries workshop.

Building awareness of how you can represent the same concept in multiple different ways will help you avoid doing so without explanation while teaching.

This exercise should take about 5 minutes.

What Problem?
Experts sometimes solve problems before even stopping to recognize that they have encountered one.

-- -- *Exercise: Diagnosis* -- --
What is an error message that you encounter frequently in your work? (These are often syntax errors.) Take a few minutes to plan out how you would explain that error message to your learners. Write the error and your explanation in the Etherpad.

-
-
-
-
-
-
-
-

This discussion should take about 5 minutes. (Optionally, this may be discussed in group breakouts, adding 5 minutes.)

“Just” and Other Dismissive Language

-- -- *Exercise: Changing Your Language* -- --
1) What other words or phrases, besides “just”, can have the same effect of dismissing the experience of finding a subject difficult or unclear?
2) Propose an alternate phrasing for one of the suggestions above.

Write your examples and alternatives in the Etherpad.

-
-
-
-
-
-
-
-

This exercise should take about 5 minutes.

"Any Questions?"

You Are Not Your Learners

- Primary goals
- Concerns about time investment

The Carpentries Is Not Computer Science

Expert Advantages

The Importance of Practice (Again)

Keypoints:

- Experts face challenges when teaching novices due to expert awareness gaps.
- Things that seem easy to us are often not experienced that way by our learners.
- With practice, we can develop skills to overcome our expert awareness gaps.

Memory and Cognitive Load

https://carpentries.github.io/instructor-training/05-memory
Questions:

- What is cognitive load and how does it affect learning?
- How can we design instruction to work with, rather than against, memory constraints?

Objectives:

- Remember the quantitative limit of human memory.
- Distinguish desirable from undesirable cognitive load.
- Evaluate cognitive load associated with a learning task.

Types of Memory

- short-term vs long-term memory
- 7±2

-- -- *Exercise: Test Your Working Memory* -- --
This website implements a short test of working memory.
https://miku.github.io/activememory/
What was your score? If you are comfortable, share your answer in the Etherpad.

- 7 -- harder than I thought!
- 9 -- after two or three attempts
- 6 -- would be more if I were to write them down and then 'chunk' them for myself (or seeing all words at once and make a sentence out of them would probably lead to almost all 20)
- 4 -- after two attemps
- 3 😂 I've accepted that my short term is the worst
- gave up can remember 6 or 7 but can spell them
- three attempts: 3, 6, and 7
- 5 in the end
- 3, 5, 4 -this was challenging. I'm always disappointed by my working memory and recall, which is why I take notes when I'm trying to learn something. I also think giving students handouts, notes, access to powerpoints is soooooo imporant after the workshop, course, training, etc.
- my attempts were 2, 4, 6
- 2 lol, nope i can't do word recall today.
- 5 i got
- 10 - word association to group them helps
- 6
- 7/5 i did twice
- 5 5 6 5, I always remember the last ones
- 3,4 ;'(not try to memory all of them, just focusing some of them (group) , skiping one-two would help i guess

If you are unable to use this activity, ask your Trainer to implement the analog or adapted version of this test.
This exercise should take about 5 minutes.

Most people will have found they only remember 5-7 words. Those who remember less may be experiencing distraction, fatigue, or (as we will learn shortly) “cognitive overload.” Those who remember more are almost invariably deploying a *memory management strategy*.

Strategies For Memory Management

Chunking

Image: A list of words: cat, apple, ball, tree, square, head, house, door, box, car, king, hammer, milk, fish, book, tape, arrow, flower, key, shoe. Underneath, the same words are organized into boxes with related terms e.g. cat fish milk ball and apple flower tree"
https://carpentries.github.io/instructor-training/fig/chunking.svg

-- -- *Exercise: Improving Short-term Memory with Chunking* -- --
Repeat the memory exercise you did earlier, but this time, try to form short stories or phrases, or a visual image, from the words you see.
Write the number of words you remembered in the Etherpad. How does this compare with your first attempt?

-
-
-
-
-

This exercise should take about 5 minutes.

Using Formative Assessment to Support Memory Consolidation

Frequency of assessment

Group Work

Opportunities for Reflection

Limit Concepts

Attention is a Limited Resource: Cognitive Load

3 types:

- Things they have to think about in order to *perform a task* (“intrinsic”).
- Mental effort required to *connect the task* to new and old information (“germane”).
- *Distractions* and other mental effort not directly related to performing or learning from the task (“extraneous”).

Is Guided Practice "Hand Holding"?

-- -- *Exercise: Mapping Cognitive Load* -- --
Look in the curriculum that you chose to prepare for this workshop and focus on one step or task that

learners will be asked to complete.

- What concepts will learners need to understand and hold in short-term memory in order to complete this task?

I chose Starting with Data in Python: learners will need to keep in short-term memory the common methods used to interface with pandas dataframes.
for writing their first query in SQL: learners have to remember two new SQL statements, as well as syntax that follows and names of columns in the example data table
downloading the tool, understanding how to install the software and where data was downloaded to and where it lives on their computer,With OpenRefine learners work from a data set they download from the lesson materials and will upload to create a project.
I chose the pandas task. In order to complete this task, learners have to know about the Jupyter execution procedure.
I chose importing/reading files in R and it requires them understanding file paths, the lines of code to read in as well as appropriate libraries if necessary and how to view column headers
having R and RStudio installed and the environment prepared BEFORE the episode (Data Analysis and Visualisation in R for Ecologists/Starting with data): then, they need to have understood what an 'assignment' is in R, where there data is located on their own computer, perhaps also how to set and change directories in R
I chose the shell lesson. The very first episode of "Working with files and directories". In this case there are just some things that need to be kept in memory (cd, ls, pwd etc.), but they will be extensively repeated afterwards. So as long as that is linked and repeated, it is completely fine. Moreover, it helps to explain what cd (for example) stands for, so someone can link/associate that with the full name (that could be helpful for learners).
How to install packages and the different interfaces to interact with code, R, and the IDE (R for Reproducible Scientific Analysis)
I choose Unix Shell lesson and the fist episode would be basic commands. To practice of on a local computer with the preperad hands on session for those bacis commands.
Understanding the basics of IoT and how it relates to components to get information
To create a basic visualization, they need to know how to install libraries, how to create a simple plot in R, and later how to edit. I'd say that they need to know very few things, also order of execution is important.
I chose the introduction to python and the things that students will need to hold in memory for the short term include; wrting a script, defining variables, differents between lists/arrays and how to index, writing functions, loops. Lastly, if they have issues what steps to try and fix the issue (debugging).
Getting the data, how to load it into the software of choice *python/R , how computers work holding data in memory/disk and how its accessed
Data in R can be represented in different forms/objects and types that come with different characteristics. Knowing which data type you're dealing with is necessary for data wrangling (Starting with Data in R)
Faceting in Open Refine, some things a learner will have to keep in mind while performing this task: Open Refine is a tool for working with data sets, the data set you are working with is in a spreadsheet with rows (records) and columns (values), a Facet groups values in a column, you can filter data with these Facets, you can edit values across many rows (records) with Facets.
Data directory, Python libruary pandas. The task is to import a cvs file into Python.
R for Reproducible Scientific Analysis > Exploring Data Frames > Challenge 1: Learners will need to (1) know how to construct a vector (2) and a list, (3) as well as remember that columns hold elements of the same type and rows may hold elements of different type.
When doing git, I would focus on the concept of what is "a change" is and how to orient yourself. What is a change, where did it come from, and where you will apply it a key way of reasoning about the

complexity of version control I use in my head.

- Draw a concept map connecting these concepts. What relationships do learners need to understand to connect them?
- How many of these concepts and relationships have been introduced since the previous step or exercise?

With a partner or in small groups, discuss what you have found. Are your learners at risk of cognitive overload at this point in your workshop? Why or why not?
This exercise should take about 15 minutes.

Attention Management in Your Workshop

Using Formative Assessments for Memory Management
There are many different types of exercises that can focus attention narrowly and help to avoid cognitive overload. Carefully targeted multiple choice questions can play this role. A few more that you may wish to consider are:

- Faded examples: worked examples with targeted details "faded" out – essentially fill-in-the-blank programming blocks
- Parson's Problems: out-of-order code selection & sorting challenges
- Labelling diagrams or flow charts (may also be organized as a fill-in-the-blank)

What to Display

Keypoints:

- Most adults can store only a few items in short-term memory for a few seconds before they lose them again.
- Things seen together are remembered (or mis-remembered) in chunks.
- Cognitive load should be managed through guided practice to facilitate learning and prevent overload.
- Formative assessments can help to consolidate learning in long-term memory.

Building Skill With Feedback

https://carpentries.github.io/instructor-training/06-feedback
Questions:
How can I get feedback from learners?
How can I use this feedback to improve my teaching?
Objectives:
Describe three feedback mechanisms used in Carpentries workshops.
Give feedback to your instructors.

Surveys
For links to our surveys see: https://carpentries.github.io/instructor-training/06-feedback#surveys

The survey links above are only for you to preview the survey as part of Instructor Training. When you are teaching a workshop, make sure to share the links generated on your workshop website. Doing so will ensure that you will receive all the survey results from your workshop participants.

Image: Screenshot of a workshop website showing location of customized survey links
https://carpentries.github.io/instructor-training/fig/surveyscreenshot3.svg

Timing matters

Minute Cards

Example positive prompts:

- One thing you liked about this section of the workshop
- The most important thing you learned today
- A new skill, command, or technique you are most excited about using

Example constructive prompts:

- One thing you did not like or would change about this section of the workshop
- One thing that is confusing / you would like clarification on.
- One question you have

Be Explicit About Using Feedback

One-Up, One-Down

-- -- *Exercise: Give Us Feedback* -- --
Write one thing you learned this morning that you found useful on one of your sticky notes, and one question you have about the material on the other. Do *not* put your name on the notes: this is meant to be anonymous feedback. Add your notes to the pile by the door as you leave for lunch.

Key Points

- Give your learners time to fill out the post-workshop survey at the end of your workshop.
- Take the time to respond to your learners' feedback.

https://forms.gle/CeuYJ43mVrxtfUpK9

Question for next session: where did the term "episode" come from in Carpentries curricula? I noticed instructors used this early in the course, but was unsure the rationale or reasoning, is there one?

- The Collaborative Lesson Development Training materials come to the rescue!
- https://carpentries.github.io/lesson-development-training/episodes.html

(Rob asked me to leave this question here for the next session.)

END OF DAY ONE

Motivation and Demotivation

https://carpentries.github.io/instructor-training/08-motivation
Questions:

- Why is motivation important?
- How can we create a motivating environment for learners?

Objectives:

- Identify authentic tasks and explain why teaching using them is important.
- Develop strategies to avoid demotivating learners.
- Distinguish praise based on the type of mindset it promotes.

Motivation Matters

- Any technique can fall flat when learners are not motivated
- In a short workshop, motivation to continue learning independently is a critical outcome

How Can Content Influence Motivation?

- Believing that something will be too hard to learn often becomes a self-fulfilling prophecy.

Image: A stylized graph with y-axis labeled "usefulness once mastered" and and x-axis labeled "mean time to master". The upper left quadrant says "teach this first" and the lower right quadrant says "do not bother". https://carpentries.github.io/instructor-training/fig/what-to-teach.png

-- -- *Exercise: Authentic Tasks: Think, Pair, Share* -- --

1) Think about some task you did this week that uses one or more of the skills we teach, (e.g. wrote a function, bulk downloaded data, built a plot in R, forked a repo) and explain how you would use it (or a simplified version of it) as an exercise or example in class.
2) Pair up with your neighbor and decide where this exercise fits on a graph of “short/long time to master” and “low/high usefulness”.
3) In the class Etherpad, share the task and where it fits on the graph. As a group, we will discuss how these relate back to our “teach most immediately useful first” approach.
 -Task: looking at a spreadsheet of registration data and drilling down with filters and facets to analyze who is will be attending an upcoming training, their affliations, and such. Cleaning up any data to import registrants into a course mangement system.

-We discuss that a technical task like showing how to open a file in a terminal, shouldn't be intimidation for our students, therefore it is better to show how to ended up the task being clear and specific, only showing that specific activities in isolation.
- Task: Turning static student exercises into dynamic web pages (using Shiny and Quarto); Mean time to master: takes longer; usefulness: medium (we could do without, but it's nice to have)
- Task: teach python/R code to students. Figure out the logistic flow between jupyter notebook cells: short time to master; high usefulness; plot interaction in R using interaction.plot() instead of ggplot: short time to master; high usefulness.
-One of the tasks we talked about was using Excel Forecasting with datasets and we determined that while the skill itself is easy to learn at first it still takes a long time to master because it requires inherent understanding of data/statistics and what you're working with but it does have high usefulness.
- Automating tasks is something that we both do regularly. However, this can be a bit much and not that useful for people wanting to analyze just one data set. So we would categorize it as content that is not as useful right now and tasks more time to master. So something that should not be tought immediately.
- Learning to code python: Python interpeter, can take a while to master using but then becomes useless once mastered as you may fine a different editor you perfer.
-Task: using an API to pull data into R and begin building a dataset. This would take a medium time to understand/ master but short time to learn with high usefulness (if they are needing data from an API, of course)
- 1) Migrating reporting from one ILS to a new one used by a university and doing it with OpenRefine. 2) Forking code and synchronizing changes, I admitted the challenge of this it has a fun binary distribution: it either has a usefulness you need or you do not need it and likely a group of developers avoids for fear of complexity.
We discussed how teaching students the basics of python and R in handeling data. We also discussed the basics of VR and datavisulization for students and academics.
-One tecnique I used is to apply simplied real life data in assignments. Immeidiately useful first. My partner talked about developing an interactive tool for statistics, which is very useful to motivate students.
Task: setting up VSCode with GitHub local branches as a worktree, and working on multiple branches locally. Immediately useful on how to setup environments for working with GitHub and Code.
- Task:
Loading .csv data into Pandas, either via downloading into project directory and calling read_csv or feeding the URL to read_csv. This is immediately useful, and probably the first thing a student needs to know how to do if they're working with Pandas.
-Task: Teaching bioinformatics students (undergraduates) basic molecular biology skills (e.g. DNA/RNA extraction, PCR,...) - they most likely will never use this particular technique in their own daily life, but it should help them to later plan their own experiments and to have an idea how long a task takes, what it involves, including, for example, field work, the existence of a molecular lab, legal requirements...
we talked about cleaning up data in OpenRefine, using Git for file sharing
We discussed about our background and what type of works we are doing than our teaching experieneces.
Task: Python dataframe and Unix shell emacs vim editors.
This exercise should take about 10 minutes.

How Can You Affect Motivation?

-- -- *Exercise: Brainstorming Motivational Impacts* -- --

Think back to courses you have taken in the past and consider things that an instructor has said or done that you found either motivating or demotivating. Try to think of one example in each case, and share your example under the appropriate heading in the Etherpad.
This exercise should take about 5 minutes.

Motivating experiences

- After a poor performance from the class on a test in undergrad, my instructor prepared his "failure resume", a list of setbacks and shortcomings he experienced. Instead of focusing on how bad the performance was, he shared his "failures" as an example of how missteps and bad grades don't need to affect the rest of the course. We students formed a class-wide study group and were driven to show him we could 'bounce back', and it worked.
- when have completed the required work, the instructor responding to how to go beyond the outlined information and show a genuine interest in what has been done
- After a terrible performance in the first part of one of the course that I have, the professor asked me the reason, and offer the chance to present the work in other programming language. That allow me to show my understanding of the models instead on focus only in my challenges.
- My professor has been super awesome in allowing me to work with datasets I actually have to work with on a regular basis for my practical application assignments because I expressed I was having a hard time connecting with the provided dataset to use which has exponentially improved my ability to grasp the concepts we're learning about and makes me think of how to apply what we're learning in what I'm doing
- A professor praised my second project of the year and pointed out ways in which it was a big improvement from the prior project. It made me feel like I was moving in the right direction
- when the instructor provided a task with an open outcome and in which they participated
- The instructor said the project we proposed would have the chance to run it using his funding.
- In a college Chinese course, singing a nursery rhyme as a 19 year old college student as part of a mid-term exam
- Motivating "Next week we will make a lab visit and see how this theory is applied".
- When the instructor provides individual support even though others have moved on during the excercise.
- When the professor allows you to come to a deadline with imperfect and not entirely complete work but where you are also able to show that you gave a concerted effort to solve the problem; things may not be clicking and this allows the professor could provide additional guidance.
- my PhD supervisor, when to the world seemed to be burning around me after making a massive error, causing months of research to be useless. Supervisor "we still have time dont we and please tell me your not giving up yet"
- "This can be useful not only in [x], but [y] and [z]....."
- Had an instructor who talked about the ethical implications of metadata work, not just the precision and correctness we strive for.
- Had an instructor allow for redos on all work, can't remember the name of the grading method, but you could redo work until you got perfect (or the grade you wanted/needed)
- Instructors providing additional resources and excersises, when course material has already been completed. Basically offering opportunities for faster students to go one step further on their own, while the instructor can focus on guiding slower learners.
- My instructor told a very interesting story to motivate the topic.

Demotivating experiences

- See above, singing a nursery rhyme. (This one cut both ways for me)

- "You will most likely never use this bit of code again, but..."
- "You probably don't need to know this, but..."

* motivating: next week we will make a l

- Learned python from an instructor who spent most of the class just talking through the homework from the previous class, it was extremely boring. Also, I'm a librarian and none of the examples in the course applied to my industry or needs for python.
- Demotivating: Here's the 10 pages long notebook, go through it step by step the next hour. Felt overwhelming!
- One lecture was spent learning how to use the map() in the purrr package in R, which conceptually is difficult to grasp especially for novice and intermediate learners. When one learner asked to show more examples in the following lecture to walk through it again with the whole class and see it in practice, the instructor advised that learner to use the resources provided to practice on their own rather than taking additional class time (not exactly his words but this was the message that was conveyed)
- taking for granted what we should know and not listening to feedback that we dont know something.

- Learning GIS online my instructor provided a syllabus, but did not provide a primer with basic functionality using the software.

- Being told my confience in what I knew already made me overlook how much more I didn't know/ had yet to learn
- I've had a handful of times where I'd be in a class where we're supposed to use R, or RapidMiner or whatever system (this was in the way beginning of me learning the systems or any coding at all) but the professor had no knowledge on how to use the software, the institution offered no tutoring specific to the software we were expected to use, and basically just being told 'try your best'. Fortunately I was working for a Coding Bootcamp at the same time so I had a great crew of colleagues that were able to help
- My chemistry lecturer once said in class that its useless to have a chemistry degree beacuse chances are, we will not be employed, I am now doing Physics
- Instructors not being able to clearly outline what the purpose of an exercise is or what the information/knowledge is useful for.
- I gave an answer to an instructor's question, and he responded "Now see guys, you're in graduate school, you can't give stupid answers like that anymore".
- "You should not ask learners what they want as you might get way too many answers"
- When an instructor indicated that he will not have any of the discussions in English but another language and his explination was that the coursework was in english.
- Heres the long way to do it, but most of the time just use this. Actually showing students a contents page beforhand.

Invite Participation

- Establishing norms for interaction
- Encouraging learners to learn from each other
- Acknowledging when learners are confused

Encourage a Growth Mindset

- Positive error framing

-- -- *Exercise: Helping Learners Learn From Mistakes* -- --

A learner at your workshop asks for your help with an exercise and shows you their attempt at solving it. You see they’ve made an error that shows they misunderstand something fundamental about the lesson (for example, in the shell lesson, they forgot to put a space between ls and the name of the directory they are looking at). What would you say to the learner?
In the Etherpad, describe the error your learner has made and how you would respond.
This exercise should take about 5 minutes.

- I would be careful to assume that the learner really has made a misconception - could also just be a typo and difficulties to find that typo (I am thinking of people with some forms of legasthenia - might not be the correct word in English). So, to get around that problem I would first ask the learner to repeat in words what they wanted to achieve and then let them type in again - if still wrong, I would let them make a comparison with the lecture notes to figure out themselves about the missing space
- Speak about the shell being very sensitive to keyboard strokes, and tell them that typos will happen, encourage them to keep using the sticky pad and asking helpers for support, because we are all making mistakes together!
- I would say that before the return it will help to double check the command on the screen will help more attention on the exercies. Because it is common typo errors in Unix shell.
- I might frame it by sharing common mistakes I look for when I get an error and see if they can spot it (i.e. "I usually look for extra spaces and types when I see errors or gen unexpected results")
- I know I wouldn't say: "Just put a space between [x] and [y]".
- I would ask them what they think the error is, encouraging them to take into account the error message they get
- I would try to take a pair-programming style approach and encourage us to look through the lines of code together to identify where the issue may be - if the learner realizes the mistype when we get to that line then we're groovy but if not I would go over that the space would be needed (I'm notorious for mixing up my })] or missing a comma and make a lighthearted joke about how it can be difficult but if unsure to try different options if necessary to refresh their memory
- I would say it seems that you may have mistyped something in the shell; review your code and make sure the program knows what you are trying to call and remember how the end of calls are denoted in the shell
- I would ask the person to check their input and to see if it is correct. If they say it is correct, I would explain why you would need a space between the ls command and directory. I would also ask the person to try another example.
- I would be clear: I see a space is missing between between Is and name the name of your directory, that is probably going to cause an error message. I would also try to find something positive to say about their work, because that space is probably just a small piece of a much large body of well done work.
- Using the shell typo mistake, I'd ask the student to first read what they typed and ask if they can make sense of the error message. If they catch it themselves after being given that push, great, if not, I'd help them understand that the error message implies there was no valid 'ls' command entered. With some gentle prodding, I'd nudge them towards realizing they forgot a space.
- I think that it seems that we need to start from the beginning and I will make a short intro about the concept and I will show how to solve the problem.
- I would say:"I would do like this,"(remove the space for him) "does this work now?"

* I would say: Oh, this is a mistake I also always did in the beginng, maybe you can pass this line into ChatGPT and see it's suggestion

- Presenting the Instructor as a Learner
    - "The typos are the pedagogy"

- Praising effort or improvement, not performance or ability
- In this example the error usually has output. I get excited for errors and point out they are learning opportunities. I start by asking for intuition about what the error message means and how would they solve it. Depending on time I give them one or more attempts to find a resolution and step in once asked.
- "You missed a space there after the ls" then normally follow up a with a few questions to narrow down what they have miss understood and would then explain that aspect (targeted apporach). youve got to watch that you dont expain the wrong thing as it short cut to causes some great levels of confusion.

-- -- *Exercise: Choosing our Praises* -- --
Since we are so used to being praised for our performance, it can be challenging to change the way we praise our learners. Which of these examples of praise do you think are based on performance, effort, or improvement?

- That's exactly how you do it – you haven't gotten it right yet, but you've tried two different strategies to solve that problem. Keep it up!xxx
- You're getting to be really good at that. See how it pays to keep at it?xxx
- Wow, you did that perfectly without any help. Have you thought about taking more computing classes?
- That was a hard problem. You didn't get the right answer, but look at what you learned trying to solve it!xxx
- Look at that - you're a natural!

This exercise should take about 5 minutes.

- Leveraging the power of "yet"

First, Do No Harm!

Things not to do in a workshop:

- Talk contemptuously or with scorn about any tool or practice, or the people who use them.
- Dive into complex or detailed technical discussion with the one or two people in the audience who clearly don't actually need to be there.
- Pretend to know more than you do.
- Use the J word ("just") or other demotivating words we talked about in a previous lesson.
- Take over the learner's keyboard.
- Express surprise at unawareness.

Not Just Learners

- Why does your motivation matter?

-- -- *Exercise: Why Do You Teach?* -- --
We all have a different motivation for teaching, and that is a really good thing! The Carpentries wants instructors with diverse backgrounds because you each bring something unique to our community.

What motivates you to teach? Write a short explanation of what motivates you to teach. Save this as part of your teaching philosophy for future reference.

This exercise should take about 5 minutes.

Keypoints:

- A positive learning environment helps people concentrate on learning.
- People learn best when they see the utility in what they're learning and believe it can be accomplished with reasonable effort.
- Encouraging participation and embracing errors helps learners to stay motivated.

Equity, Inclusion, and Accessibility

https://carpentries.github.io/instructor-training/09-eia
Questions:

- Why are equity, inclusion, and accessibility important?
- What can I do enhance equity, inclusion, and accessibility in my workshop?

Objectives:

- Identify instructional strategies that are consistent with universal design.
- Recognize systemic factors that can distract and demotivate learners.
- Understand the role of The Carpentries Code of Conduct in maintaining an explicitly inclusive environment.

A Positive Environment for All

Definitions

- Equity: The proportional distribution of desirable outcomes across groups. Sometimes confused with equality, equity refers to outcomes while equality connotes equal treatment.
- Inclusion: Actively engaging traditionally excluded individuals and/or groups in processes, activities and decisions in a way that shares power. Inclusion promotes broad engagement, shared participation, and advances authentic sense of belonging through safe, positive, and nurturing environments.
- Accessibility: Refers to the intentional design or redesign of technology, policies, products, and services (to name a few) that increase one’s ability to use, access, and obtain the respective item. Each person is afforded the opportunity to acquire the same information, engage in the same interactions, and enjoy the same services in an equally effective and equally integrated manner, with substantially equivalent ease of use.

The Carpentries Core Values

-- -- *Exercise: Discuss The Carpentries Core Values* -- --

- Take a moment to read through the Core Values on this page: https://carpentries.org/values/
- Choose one core value that resonates with you. What is a decision you might make in a workshop that could look different if you were actively considering the core value you chose?

This exercise should take about 5 minutes.

Strength through Diversity: here I am thinking about diversity in the sense of a wide variety of background knowledge people might have in my class/workshop. I often times have almost a 'bimodal distribution' of absolute starters and rather advanced beginners - here, I could use the abilities of both to invest more time in techniques like paired programming

Accessibility

-- -- *Exercise: What Happens When Accessibility is an Issue?* -- --
Think of a time when you have been affected by, or noticed someone else being affected by barriers to accessibility. This may have been at a conference you attended where the elevator was out of service, or maybe a class you were taking relied on audio delivery of content. Describe what happened, how it impacted your (or someone else's) ability to be involved and what could have been done to provide better accessibility in this case.
This exercise should take about 5 minutes.

I only found out during the class that I had a blind student who actually could not see some of the examples (was an beginner's course in R with some visualization, ggplot etc. - I have to admit that I found it tricky to find work arounds ad hoc)
I don't know if this goes in line with accessibility, but rather different learning methods. It is usually very difficult to focus on recorded lectures even with CC & higher speed. Typically lecture transcripts are provided that make it more conducive for me to follow along but there was one week where the transcript wasn't provided/included and I definitely could tell how it impacted my ability to learn and understand that week's topic
Whenever I am teaching a coding workshop and someone has a apple mac.....(not always but majorty of the time) they always end missing out on the first few exercises due to technical issues with permissions. Its got to the point where I now borrow 5 laptops from the university everytime I teach to fix this issue.
Internet stops working during an online class. I upload recordings to back up for this situation.
One of the building in the campus doesnt have an elevator that allow to lift a wheelchair, so part of test were taken in First Floor with only one room for test. So, one person in my cohort needs to request ahead of time when she was planning to take a test.
I was trying to find a quite room to pray when i was rushing a traning session.
I can't readily recall a barrier that I experienced or where I witnessed someone else's experience but I did recently attend a conference where a speaker discussed the barriers to women, specifically pregnant women and single moms in the STEM field in terms of accessibility to privacy rooms or even into entering the field after maternity leave, and that really resonated with me.
I attended a conference that was intentionally convened in the agency's government office and remote access was discouraged, where many key developers were not government employees, U.S. citizens (so access was onerous and trips too expensive to consider), or both. The training workshop was not remotely

accessible by video, so people who need the training the most could not access it.
Open day at a university - a person in a wheelchair could not visit the library due to the lift being broken. The person was still in high school and I doubt she applied to study at that university.
Had a student join an online class meeting from her local library, she didn't have reliable space at her home to join. She contacted me before the class meeting, to let me know that she would be late because the library opened when our class meeting started. I started the class material a little later (by extending the student "coffee chat") at the beginging to make sure she was able to join us.
I recently when to a conference without shared presentation materials, it was hard to take notes/ take in information with the speed of the presentations (specially image heavy slides)
We had an exchange student who was the only one not speaking German. Lots of discussions in the workshop happened in German, so for him it was difficult to get involved.
We have a 'diversity hour' called 'Beer hour' (you guess why) regularily on Thursdays at 5 pm. People there celebrate how diverse their group is in terms of gender, culture, religion, countries, languages... However, you may start wondering why there are still some people not participating. You may think about parents with small kids, people from various religions, pregnant women... It is sometimes hard to imagine a barrier that I experienced or I witnessed someone else might experience if the person who might be affected is actually not even there
From Accomodation to Universal Design

Image: Cartoon showing strollers, suitcases, bicycles, carts, and wheelchairs using curb cuts
https://carpentries.github.io/instructor-training/fig/sketchplanations-the-curb-cut-effect.png

Universal Design in Learning (UDL)
The key to UDL is creating redundancies such that learners have multiple options in how they:
1) receive
2) engage, and
3) share information.

-- -- *Exercise: Activity: Applying Universal Design in Your Teaching* -- --
Consider some of the teaching tools and strategies we have discussed so far in this workshop, or others you have observed in your experience. How do these meet UDL goals of providing multiple options for learners?
Consider multiple ways for learners to:

- receive information
- engage with you, the material, and other learners
- share what they have learned

This exercise should take about 10 minutes.

Every Little Bit Counts

Accessibility Testing

Systemic Exclusion

Stereotypes

- may be explicit (conscious and deliberate) or implicit (unconscious and automatic)
- guide what we notice about people
- guide how we interpret people's behaviors
- can facilitate quick judgements in appropriate situations (e.g. stopping a child from driving a car)
- can lead to systematically negative attitudes and behaviors towards members of certain groups

When Instructors have stereotypes about learners
When learners experience stereotypes about themselves

What can we do about our own stereotypes?

Better Together: Learning with Friends

Equity versus Equality

Inclusive Practices in a Carpentries Workshop

Setting Expectations with the Code of Conduct

Listening with Assessment and Feedback

Examining your Actions

Looking for More? Want to Contribute?

The Carpentries is actively working on improving our content and practices with respect to equity, inclusion, and accessibility. If you are interested in being involved in the development of this content, please let us know! Contributions to this page may be made on GitHub (click the "improve this page" link at the top), though our #accessibility channel on The Carpentries Slack, or by emailing team@carpentries.org.

Keypoints:

- Inclusivity is a key attribute of a positive learning environment.
- Universal design benefits everyone.

BREAK (15 min)

Teaching is a Skill

https://carpentries.github.io/instructor-training/11-practice-teaching
Questions:

- How can I improve my teaching?

Objectives:

- Use peer-to-peer lesson practice to transform your instruction.
- Give thoughtful and useful feedback.
- Incorporate feedback into your teaching practices.

Lesson Study: Applying a Growth Mindset to Teaching

Jugyokenkyu or "lesson study": the power of classroom observation

Reading It Is Not Enough

Feedback Is Hard

Image: A three panel comic. In the first panel, a smiling figure is surrounded by speech bubbles with mostly positive feedback. In the second panel, the figure is eating dinner. All of the previous speech bubbles appear faded out, except the one negative bubble. The third panel shows the figure in bed, with an unhappy face, with the one piece of negative feedback lingering after all others have faded.
https://carpentries.github.io/instructor-training/fig/deathbulge-jerk.jpg

- Initiate feedback
- Be specific
- Balance positive and negative feedback
- Provide a clear next step
- Communicate expectations
- Remember that giving feedback is a skill
- Use a feedback translator

Lunar Babboon comic: https://web.archive.org/web/20210513225525/http://www.lunarbaboon.com/comics/feedback.html

-- -- *Exercise: Giving Feedback* -- --
We will start by observing some examples of teaching and providing some feedback.

Watch this example teaching video as a group and then give feedback on it.
https://www.youtube.com/watch?v=-ApVt04rB4U Put your feedback in the Etherpad. Organize your feedback along two axes: positive vs. opportunities for growth (sometimes called “negative”) and content (what was said) vs. presentation (how it was said).

Note: there is a version of this video with subtitles in both Spanish and English here:
https://www.youtube.com/watch?v=jxgMVwQamO0

This exercise should take about 10 minutes.
Positive
- Content

- excellent hand actions, shows intress in eagerness
- includes a "have to fix something" as going through to show that it's normal to have to correct
- live coding
- running an live example
- seems to have a great understanding of the material and presents many examples
- explains the utility of 'def()' function (if I heard correctly) and its gaps
- Honest when he made a mistake
- good speaking speed, makes a pause to ask for questions
- task-oriented with instruction
- examples were modular or self-contained
- live codes and show typos can happen
- relating the code to a function in excel could have been a useful comparison
- Tries to explain at the end what they want out of the exercise and how it was successfully done

- Presentation

- he actually uses live coding (which also makes it a bit slower)
- Clear speaking voice, easy to understand
- Live coding and speaks clearly.
- It is hard to see and the explanation are a bit complex.
- tends to look at the audience and speaks at a sufficient volume
- little distractions, only one active shell
- clear and very confident
- stops to check a message on his phone

Growth opportunities
- Content

- the structure of the content being taught could be more clear. Adapting variable names could be a great way to make understanding easier.
- too much wording
- there are a lot of different aspects of functions being covered
- Could pause in the middle for students to take in
- Introducing lower-level concepts before higher-level concepts would help students be on the same page. Leads to less "If you don't know what xyz is, this is what we expect"-style comments.
- Requires understanding of other concepts to be able to follow along
- Speaking fastly and ruhsing the over topics not trying to eyes contact with learners
- 3*3 + 3+3 is 12? Perhaps just too fast, but I did not get this example. And would not have dared to ask back given his presence and attitude
- he does not seem very practiced or methodical..
- he re-words his instruction often making it difficult to follow, does not re-cap the last section before they took a break, asks "any questions" at the end
- Could use more inclusive language
- usage of some technical terms is not motivated
- could be a good idea to practice the exercise a few times before giving it

- the atmosphere could be a bit more welcoming
- He can be more polite to a late learner, ask everyone including himself not to use mobile devices during a training session, expand on certain topics to help novices understand the matrial better.
- Use of short, single-letter variables

- Presentation

- rude "sit down now" welcome back from the break
- Does not seem to really like teaching in my view, seems to have the 'I want to get it over' attitude
- he spends too much time in the begining of the lesson policing students, treat folks with respect,
- text on screen is pretty small
- speaks too fast for non-native speakers I would say
- "we can just...", "this is really simple stuff...", "as you'd expect...", "don't worry about this, trust me"
- "EVEN EXCEL USERS CAN DO THIS"?! Yes.. that was great :-D
- checks phone in the middle of lecture
- attitude towards ppl standing,
- Make the content more visible, and speak more slowly when stating technical terms -- accessibility issues
- hard too watch
- should zoom in the coding area to improve readability
- very flighty, rapid
- turning back toward students, small text, lots of "um" and filler words - would be very difficult for students with poor vision or hearing
- Frequent use of the word "right?" without pausing for actual knowledge check (intention is good delivery could improve)
- Could use a pointer instead of hand gestures
- Frequent use of terminology not for novices or those not in computer science curriculum (nothing good has come from me mistakenly teaching polymorphism or analogies to it)
- Even when scaffolding, the instructor could help students by presenting more slowly.
- small font size and fast presentation speed make it hard to follow.
- own mistakes are not corrected for everybody to understand what went wrong
- Hard to see the live coding, explanations can be expanded, and can ask if everyone understands. Also explain more what he is doing while live coding and not assuming everyone understands.
- Nothing is clearly visable, attitude is a bit of an issue and the geneal flow is too fast.

-- -- *Exercise: Sharing Feedback* -- --
The prep time for this exercise is intentionally short – the point is to practice giving and receiving feedback, not to create a perfect presentation. Imperfect presentations will give you more to work with!
Distributed trainings:

- Split into groups of three.
- Individually, spend 5 minutes preparing a 90-second introduction to the topic of the lesson episode you chose before the start of the training course. You will not be live coding.
- Get together with your group and have one person teach their segment to the group. Keep a strict time limit of 90 seconds per person (one person should be responsible for the timekeeping).
- After the first person has finished teaching, share feedback. The person who performed should start by offering feedback on themselves. The timekeeper should help to keep feedback to about 5 minutes per person to ensure everyone has time to perform and discuss.
- Rotate roles and repeat steps 3 & 4

- Return to the main group and briefly summarize the feedback you received in the Etherpad. Your Trainer will split the group into virtual break-out rooms. Follow the instructions above but do not record each other. Instead, give each person feedback immediately after they finish their turn teaching.

This exercise should take about 25 minutes.
 more time needed!

-- -- *Exercise: Using Feedback* -- --
Look back at the feedback you received on your teaching. How do you feel about this feedback? Is it fair and reasonable? Do you agree with it?
The feedback I received was very helpful and I particularily liked that it was given in person (not on cards or similar) - I know that not everyone is feeling comfortable in teaching, but I personally react much more open to feedback that is given frank and in person
 The feedback received was spot on to what I had expected
Feedback was fair and pointed out a perspective that I did not consider while I was planning the introduction
My own fault that I forgot to turn on my camera, meaning that people weren't able to give feedback about body language. Otherwise yes. I believe this exercise works well in a face-to-face setting. Maybe it's just me but the 90 seconds felt a bit awkward.
The feedbacks help me take the perspective of audience that switching between screens might be distracting.
The feedback will help me to correct some misassamptions in my future presentations.
Feedback was fair and informative, good to get difference perspectives
The feedback I received was reasonable and I totally agree with it
It was a very short time limit to do a quick introduction to a topic that maybe folks have not used before. I appreciated the practice, I think we all felt that we needed to rush because of the time constraints.
Feedback was great and poignant, well constructed and presented in a warm way.
The feedback was encouraging to improve and feel comfortable to develop my teaching skills. I had some assumptions about what they suggest and now i am sure that i could be more clear for fundamentals when i do teaching.
We usually agreed on feedback and all found it helpful. It's always nice to give a suggestion on how to make it better and not just what you did not like.
I thought it was fair and reasonable. It can be hard to give constructive feedback, and my partners did will with the "compliment sandwich" concept.
The feedback was positive and provided some helpful hints for my presentation.
The feedback was kind and positive, I think I generally agreed with it.
I agree with the positive feedback about my explanation.
Identify at least one specific change you will make to your teaching based on this feedback. Describe your change in the Etherpad.
 I will slow down and can work on taking time to explain a thing and give examples and ask for participation or questions..
 Try to improve my habit of regurgitating the same information in different ways
I will continue to practice to talk slower and clearly.
I will make sure my teaching is not giving mixed messages in terms of the lesson's objective when the act of teaching is in the pursuit of accessibility/openness
ensure all topics involved are explained, and not taken for granted
I needed to explain certain aspects (which were pointed out to be more important) in more detail, basically focus more on important details first

Combine media (audio & visual) instead of focusing on just speaking. Provide succint visual accompaniment for my introduction.
I need to be even more organized. Was a bit tough to concentrate on the content, presentation and time at the same time and I feel I would quickly more and more improvise and then loose track on the time
I need to present my information more clearly and concisely with a visual component.
This feedback was helpful, also just the act of presenting in front of people was helpful. I have some ideas to develop for the larger presentation, which is good! Also, I learned how I could improve by doing this exercise with others.
I need to work on framing and pacing given the contrived example but the feedback was constructive.

This exercise should take about 5 minutes.

Keypoints:

- Like all other skills, good teaching requires practice and feedback.
- Lesson study is essential to transferring skills among teachers.
- Feedback is most effective when those involved share ground rules and expectations.

Wrap-Up and Homework for Tomorrow

https://carpentries.github.io/instructor-training/12-homework
Questions:

- What have we learned so far?
- What needs to be done to prepare for the next part of the training?

Objectives:

- Describe overnight homework.
- Produce a paragraph, drawing, or diagram that summarizes what was taught to this point.

-- -- *Exercise: Feedback* -- --
The Trainer(s) will ask for feedback on the day in some form.

Was the content taught today:
- too much:x
- too little:xxxxx
- just right:xxxxxxXxxxxx

Was the speed of the delivery:
- too fast:x
- too slow:
- just right:xxxxxxxXxxxxxxxxx

This exercise should take 5 minutes.

-- -- *Exercise: Reflection Exercise* -- --
Before we wrap up for the day, take 5 minutes to think over everything we have covered so far. On a piece of paper, write down something that captures what you want to remember about the day. The Trainers will not look at this - it is just for you.

If you do not know where to start, consider the following list for a starting point:

- draw a concept map, connecting the material
- draw pictures or a comic depicting one of the day's concepts
- write an outline of the topics we covered
- write a paragraph or "journal" entry about your experience of the training today
- write down one thing that struck you the most

This exercise should take about 10 minutes.

Keypoints:

- So far we learned about how people learn, how to build a positive classroom environment, and how to give feedback.
- Tomorrow we will cover specifics of Carpentries workshops and teaching practices.

END DAY 1 / 2nd HALF DAY

Welcome Back

https://carpentries.github.io/instructor-training/13-second-welcome
Questions:

- What have we learned so far?
- What will we focus on today?

Objectives:

- Review main points we discussed yesterday.
- Introduce topics we will discuss today.

-- -- *Exercise: Questions* -- --
Yesterday we asked you to read some resources about the logistics of teaching and running Carpentries workshops. Please add your questions about logistics and preparation to the Etherpad. We will answer these questions in the Etherpad during your work time and will return to this list later today.

This activity should take about 5 minutes.

-Will lessons (according to the Capentries concept) mostly consist of live coding (done by the instructor) and exercises (done by the participants) or are there other sections as well? E.g. how would one deal with everything that is not "code": would we prepare a slideshow for workshops in advance? Or solely rely on the live coding and the exercises?
-I think that the most relevant thing is different ideas that could be implemented in lessons.
- do the Carpentries branch out into hack-a-thon style training after course syllabus completion, I understand beyond the scope of the new learners, but has it been considered for the future? more of a problem based coding session
-Is it possible to host a hybrid centrally-organised workshop or would this have to be a self-organised workshop?
- how to accomondate students with different levels of coding skill
-What further teacher training opportunities are available to us, is there a logical sequence? Thanks
- Do you have to charge the recommended fee, or it is a recommendation? I only ask because NIST runs self-registered workshops and I am not sure federal employees can charge for events or have to follow a specific process to do so with a long lead-time.
The materials are licensed under CC-BY 4.0. To what extent can they be reused, and how to properly cite them?
Do we inform the org? when utilizing and teaching the lessons? Is there a central space to log these trainings? I have been a helper in numerous trainings, should I be logging these someplace?

Along the same lines as the accomodation question - if a potential student completes a pre-training survey and indicates that their skill is more advanced than what the workshop would provide the learner, do you turn them (for lack of better words at the moment) away and give them resources for another more 👍adanced training or contact them individually and let them know what to expect?

Keypoints:

- Instructors guide learners to construct the proper big picture (accurate mental model) of the topic rather than focus on details.
- Instructors rely on frequent feedback from learners to monitor their own presentation of the material.
- Instructors introduce a few concepts at a time to avoid cognitive overload.
- The best way to motivate learners? Show them how to do something they can immediately put to use and be enthusiastic about it.
- Teaching is a learned skill.

Checkout Process

https://carpentries.github.io/instructor-training/14-checkout
Questions:

- What do I need to do to finish certifying as a Carpentries Instructor?

Objectives:

- Describe the final steps required to qualify as an Instructor.
- Schedule your community discussion session.

Instructor Checkout

-- -- *Exercise: Be The Expert: Checkout Q & A* -- --
In small groups, read and discuss one of the three checkout procedures described on this page: https://carpentries.github.io/instructor-training/checkout Make notes in the Etherpad:

- What points do you think it is most important or helpful for people to remember?

- The choice of episode for the demo

- What questions or points of confusion do you have, or think others might have? When you are done, report back to the full group about that stage of the process.

This exercise should take about 10 minutes.
- I do not understand the 'Get involved!' step -
as it reads, it can be from 'just fix a typo in the material' up to 'be a helper in a complete workshop' - that seems to be a bit of difference. I would think the main goal would be a reasonable onboarding for us, but given all time restrictions we have in our daily lifes, I would feel that most people try to fix a typo and are done.. Did I get that right here?
- One of the important things to remember is the deadline of the checkout process before receiving the certificate. which is adequate since most of us if not all have busy schedules
- what is a "Carpentries Instructor Trainer" is this anyone who is also passed thier instructor training, or a specific select group? also do we do this as part of the instructor training (during the actually event)?
https://carpentries.org/trainers/
- can the "get involved" be done prior to the instructor training?
- deadline to complete all the required steps to get the certificate.
What subtasks include under the get involved to conclude the chekout could not get it well.
- I've read through the checkout process in the past, and found myself wondering then (and now) how to sign up for a welcome session and the other steps. I do think planning ahead for all of these is the most helpful to remember since it does take some logistics to complete every step given the individual's schedule and the carpentrie's schedule.
https://pad.carpentries.org/welcome-sessions-2024
agree + the time difference for the workshops and public holidays...
Does it matter what community sessions we sign up for? i.e. a time slot works for a community in another region than the one I am in
Does a 'Get Involved' session count (as a helper) when the session was a pilot 2 day training session?

https://pad.carpentries.org/pad-of-pads

-- -- *Exercise: Schedule a Discussion or Demo* -- --
Visit the discussion Etherpad to sign up for a session: https://pad.carpentries.org/community-sessions-2024 If the session you would like to attend is full, contact the discussion host and co-host to ask if you can attend.
If you would prefer to do your teaching demonstration before your discussion, visit the demo Etherpad and sign up there: https://pad.carpentries.org/teaching-demos This demo rubric is provided as a guide for Trainers evaluating potential new Instructors during the teaching demonstration.
This exercise should take 5 minutes.

What does a badge mean?
- teaching
- voting
- bonus modules
- sharing

Keypoints:

- To certify, you must contribute to a lesson, take part in a discussion, and do a teaching demo within 90 days of your training event.

The Carpentries: How We Operate

https://carpentries.github.io/instructor-training/15-carpentries
Questions:

- How is The Carpentries organized and run?
- What is the difference between SWC, DC, and LC workshops?
- How do you run a Carpentries workshop?

Objectives:

- Get connected with The Carpentries community.
- Describe where you can go to get information on running a workshop.

A Brief History

Global & Local Carpentries communities

Image: A very brief history of The Carpentries. A timeline. https://carpentries.github.io/instructor-training/fig/SWCDChistory.png

Similarities and Differences between The Carpentries Lesson Programs

Similarities between Data Carpentry, Library Carpentry, and Software Carpentry workshops include:

- a focus on technical skills,
- a two-day format taught by volunteer instructors, and
- a focus on filling gaps in current training for learners.

Image: Three intersecting circles labelled Software Carpentry, Data Carpentry, and Library Carpentry.
https://carpentries.github.io/instructor-training/fig/carpentries-venn-diagram_20200904.svg

What is a Carpentries Workshop? The Rules.

Using the Names and Logos
https://carpentries.org/workshops/#workshop-core

Recruiting helpers:
https://docs.carpentries.org/topic_folders/hosts_instructors/hosts_instructors_checklist.html#helper-checklist

Materials
https://carpentries.github.io/instructor-training/LICENSE.html

Reporting a 'Mix and Match' Workshop
https://amy.carpentries.org/forms/workshop/

Instructor Certification is Comprehensive

Carpentries Jargon Review
-- -- *Exercise: Test yourself!* -- --
As a class or in groups, see how many of the following terms you can define.

- Lesson
- Episode
- Workshop
- Lesson Program
- Instructor
- (Instructor) Trainer

This should take about 5 minutes.

How to Organise a Carpentries Workshop Locally

https://carpentries.org/workshops/#workshop-organising

The Carpentries Handbook: https://docs.carpentries.org/ includes:

- templates and checklists https://docs.carpentries.org/topic_folders/hosts_instructors
- policies https://docs.carpentries.org/topic_folders/policies
- much more!

Callout: Teaching Opportunities: Local and Global

Instructors mailing list: https://carpentries.topicbox.com/groups/instructors

-- -- *Exercise: Explain to a partner* -- --
With a partner, take turns asking and answering the question: “I want to organize a workshop! What will I need to do?” One partner should ask about a self-organised workshop, and the other can ask about a centrally-organised workshop. If you have a third person, they can help out with follow-up questions or answers as needed.

When you encounter new questions during this process, be sure to write them in the Etherpad.

- Is there a aggregated list of self-organized workshops in a region? I.e., a university near me hosting a self-organized workshop on XYZ, that might be within driving range for me to try and attend as a learner (assuming they permit walk-ins from outside the institution).
- https://amy.carpentries.org/

- How to make sure we correctly cite the materials from Carpentry if we do "mix"?
- https://software-carpentry.org/lessons/
- https://swcarpentry.github.io/shell-novice/
-
-

Leave about 10 minutes for this discussion.

Setting Out On Your Own… Together: Lesson Incubation

https://github.com/carpentries-incubator/proposals/

A Culture of Contribution

-- -- *Exercise: Community Roles* -- --
Select one role from the list below that interests you. Using the the descriptions on The Carpentries community website (https://carpentries.org/community/), write 1) a short definition of the role and 2) a question that you have (or that you imagine someone else might have) about the role. Are there roles you would like to see that are not listed? Note that, too!

- Executive Council
- Mentors
- Instructor Trainers
- Lesson Developers
- Code of Conduct Committee
- Instructor Development Committee
- Community Facilitators
- Maintainers
- volunteer instructor

This exercise should take about 5 minutes.

Keeping In Touch

Want to listen?

- Sign up for our newsletter
- Follow us on Twitter, Facebook, or LinkedIn

Want to interact (or listen with options to engage)?

- Join our Slack organisation
- Join our Email lists (start with “Discuss”!)

Want to join meetings (to meet new people or listen in)?

- Sign up for Community Discussions (or just drop in if there is space!) or other events when announced
- Explore taking on one of the Roles identified above

-- -- *Exercise: Get Connected* -- --
Take a couple of minutes to sign up for The Carpentries channels you want to stay involved with on this page: https://carpentries.org/connect/ When you are done, share a channel you find interesting or useful on the Etherpad.
This exercise should take about 5 minutes.

Keypoints:

- The Carpentries materials are all openly licensed, but names and logos are trademarked.
- Carpentries workshops must cover core concepts, have at least one certified Instructor, and use our pre- and post-workshop surveys.
- [Guidance for teaching and hosting workshops](https://docs.carpentries.org/topic_folders/hosts_instructors is provided in [The Carpentries Handbook](https://docs.carpentries.org).

BREAK (15 min)

Live Coding is a Skill

https://carpentries.github.io/instructor-training/17-live
Questions:

- Why do we teach programming using participatory live coding?

Objectives:

- Explain the advantages and limitations of participatory live coding.
- Summarize the key dos and do nots of participatory live coding.
- Demonstrate participatory live coding.

Why Participatory Live Coding?

Exercise: Up and Down
List some advantages and challenges of participatory live coding from both a learner's and an instructor's point of view in the Etherpad.

This discussion should take about 5 minutes.

Advantages
 - advantage: it slows down the whole instruction (which may be an advantage for both learners and instructors)

 - advantage: makes coding more approachable and helps learners see the code being build step by step.
 - challenge: potentially increases cognitive load/stress for instructor. it also takes their attention away from students (i.e. harder to see when a student appears to be struggling)
 - more engaging, gives the instructor credibility, extra element of time and execution order.
 instructor - feedback can be instant, with live coding, learner-seeing how its actually done and not just copy&paste, focus is kept and engaged.
- easier to follow along

- encouraging to see progress and results right away on your own computer as you code along
- - It's interactive so learner see how the code develops as it's been written.
- Learners get immediate hands on experience following along, can see the "process".
- Advantage: It focuses the attention of learners on the lesson because it's so immersive, they cannot multitask easily, e.g. check notifications on the phone or read something else while the lesson
- Build "muscle memory" for key tasks and operations (instructor; learner)
- Allows you to talk through the functionality of each individual piece and turn concept into practice
- Allows the learner to see how a more experienced coder might troubleshoot code errors
- advantage: shows learners exactly how it works within the coding environment
- The advantage of live coding gives the people being trained an oppertunity to follow and repeat what is being followed, thus learning by doing.
- I get so much more from live tool working, I learn ways experienced folks navigate the tool they use.
- Demonstrate how coding works, and the realistic of debug if applicable
- Learners can see the instructor dealing with their mistakes and learn from that

Disadvantages
 - challenge: it is well possible for a learner to actually think that one has understood everything but actually one only can copy/paste would is shown - you still need the time AFTER the live coding to work on your own (and you should dedicate time for this as instructor)

- parts may be way too slow and other parts way too fast; hard for a learner to keep up concentration

- Advantages: Attenders can see the how the topic can run on the real-world application and also try it by themselves.
- Disadvantages: There would be some hurdl about set up and diffrent working environments.
- disadvantage: when the program stops working and turning on/off does nothing to fix the problem
- some functions use dual spelling (i.e summarize vs summarise) which when showing live can be confusing for those that do not use similar spelling and along the same lines, based on my experience live coding has been taught with english functions so for someone that is not a native english speaker, than understanding might be slower even during the live session
- disadvantage: additional input for the learners to process
- disadvantage: extra window/content for the instructor to share during virtual course. potentially lots of switching between open windows which might be confusing
- Issues with setup in some cases can lead to falling behind, but it is great that there are helpers
- requires the instructor to keep the lesson plan in their head
- Dis: Can cause confusion in students, if earlier steps are missed due to being late.
- environment set-up can be a challenge for the learners
- Individual may be uncomfortable with social dynamics of "pairing" inherent to this live coding (instructor; learner)
- Different cultural backgrounds may lead to discomfort at first or overtime, detracting from focus and ability to live code (instructor; learner)
- May not be able to show roadmaps of concepts
- The learners might feel rushed, could fall behind, might not have time for questions if they're busy programming.
- Having a computer at hand is very challenging for both instructors and learners (distracting emails, google, seqanswers,....)
- Making mistakes could be challenging for learning because interrumps the flow.
- not easy to keep eye contact while typing
- Making mistakes and having to rewrite the code which could confuse some learners.
- not everyone speaks up when they have problems or questions so hard sometimes to get engagement
- I personally tend to loose the 'big picture' - I might, for example, be able to copy the code how to access a server, but do not understand what I have to change if the server is different
- Can be a challenge when something does not work but we just need to be flexable to correct an issue we encounter on the fly.
- Disadvantage: Requires team teaching and coordination between instructors/helpers, because so much can go wrong with such immersive, technology rich learning experiences

-- -- *Exercise: Compare and Contrast* -- --
Watch this first participatory live coding demo video: https://youtu.be/bXxBeNkKmJE and this second demo video: https://youtu.be/SkPmwe_WjeY as a group and then summarize your feedback on both in the Etherpad. Use the 2x2 rubric for feedback we discussed earlier.

In the videos, the bash shell for loop is taught, and it is assumed learners are familiar with how to use a variable, the head command and the content of the basilisk.dat unicorn.dat files.

Note: Sometime sounds in the room can be poor. Turning on closed captioning by pressing the cc button will improve the accessibility of these videos.

This exercise and discussion should take about 15 minutes.

First video:

Content

- Positive

- nice and engaging voice and vibes
-Use comparison to demonstrate the key points
-His explanations are clear
They understood the content they were teaching

- Clear examples
- They gave multiple examples and explained what the code would do.
- Shows another way of getting the same result
- Gives examples

- Constructive

own mistake could be explained
-If waiting until the end to explain all the code, it would be good to somehow signal which line of code he is explaining so that the learner can follow along
-could have explained the slight mishap he had to get the code running correctly
background distracting on screen, notifications off ie audio, long silences instead of explaining, larger font (I second all this)
Could have engaged with the folks in the room
typed commands could be elaborated
- no explanation of what's been written or why it's done that way and no eye contact with learners

- It would be good to explain what each part of the command is doing and why it is neccessary as opposed to typing it out completely and just showing the result.
- Typing out and explaining each command will help learners follow.

Delivery

- Positive

- The pace was good and clear language
- legible font
-clear voice and good speed
- Voice is clear
Positive energy while speaking
Good pacing, slow and easy to follow
he did a good job in explaining what the code was doing (it seemed that way at least not familiar with Unix)
Use body languages
moves around more in the second video to point at the screen, making it more engaging

- Constructive

- notifications not turned off
- could go into fullscreen

-explanation only given afterwards often
- Seems to miss a question from a student (red sticky note). Could pause and check for understanding from students while explaining what he's writing.
- Fast
-Show his full screen instead of the coding area
- notifications and news in the background are disruptive
I'm in a different boat, where I think his speed of speech was actually quite fast and not very clear. especially since he waited until after he ran all of his code to explain things so I was trying to read the code he was explaining while following along to his descrption
Did not look up to see what students were communcating nonverbally
say outloud what your doing.

Second video:

Content

- Positive

engaged with the audience -
- white background works better on Beamer
- Does a good job describing the structural significance of the for loop, and how the written code is parsed
- Good explanation of the code
like the explanation of the $ to the > signs
good understandable explanation
explanation step by step
explained each component of the loop and it's code
liked that he explained another way to write the code in one line rather than in multiple lines
I like the examination of what the code did after he runs the code.
explained each part of the code, explained mistakes he made

- Constructive

-provided good explanations and pointed out typos
- 'do not do this, trust me' - seems to be very arrogant to me..
-could reiterate what the head -n 3 code is doing rather than saying this was applied to the first dataset and then looped and applied to the second
-could explain the necessity for `do` and `done` in unixshell

Delivery

- Positive
- eye contact with audience,

-really like how he is speaking as he is typing and explaining while pointing at the specific line
-is aware of red stickies
takes the time and explains very clearly
white background much better readable than black background
Standing up is more engaging (at least when you are able to stand apparently...)
-speaks out what they are typing
aware of audience potential issues, standing in front of learners, color on font removed which is clearer, is larger also. Engages by pointing physically to where the code is, and what hes talking about at the moment, so explains very well - speaking up and out to the learners, not down to his laptop in general.

- Standing up, clearly engaged with the audience, and talks contemporaneously about the code as he's writing it, pausing for reflection.
-takes their time to explain the steps
Speak in a normal pace and it is clear.
-Standing up and engaging and good pace
gestures to the screen
Engaging with the audience and checking for understanding constantly
Made eye contact, stood up and pointed to the screen.

- Makes eye contact with the participants and asks for feedback.

- Constructive
- use the technology, a large curser could be helpful as a pointer too.

-Probably use a pointer (can be difficult with colour blindness sometimes with laser pointers)
-still prefer no background to be visible even if its a standard desktop, waste of screen space
- Could ask for feedback during coding as well. "are you following along?"
- Can ask the students if they see the screen easily otherwise could make it a bit bigger
 - Could use a pointer instead of fingers
 - Users could be invited to actually give the solution themselves and not only type what they have seen (but that somehow seems to be a Carpentries policy..)
 Rather than "trust me I've been there" could rephrase somehow (agreed)

Top Ten Tips for Participatory Live Coding in a Workshop

1) Stand up and move around the room if possible. This makes the experience more interactive and less monotonous. Use a microphone if one is available to make it easier for people with hearing difficulties to hear you.
2) Go slowly.
3) Mirror your learner's environment.
4) Use your screen wisely.
5) Use illustrations
6) Turn off notifications
7)Stick to the lesson material.
8) Leave no learner behind.
9) Embrace mistakes.
10) Have fun!

-- -- *Exercise: Practice Teaching* -- --

- Split into groups of three.
- Assign roles, which will rotate: presenter, timekeeper, note-taker.
- Have each group member teach 3 minutes of your chosen lesson episode using live coding. For this exercise, your peers will not "code-along." Before you begin, briefly describe what you will be teaching and what has been learned previously. Do not record this exercise.
- After each person finishes, each group member should share feedback (starting with themselves) using the same 2x2 rubric as yesterday. The timekeeper should keep feedback discussion to about 1 minute per person; this may leave some time at the end for general discussion. The note-taker should record feedback in the Etherpad.

- Trade off roles.

This exercise should take about 25 minutes.

Thoughts and questions

-
-Using only a laptop (without printed notes etc.) is too stressful for me; it is a bit different when I am giving in-person workshops as you usually have an additional projector (exactly the RStudio impression Anne just mentionned)
-Running the code on an locked down work laptop takes too long, longer than expected, I would prepare better having run through all the coding beforehand
- Q: In this training or more generally, if we are to screenshare over Zoom and not in person, should side-by-side with website content and shell be presented that way or is there guidance? (I am sorry Florian just commented on this and someone else was typing the same thing, and I wasn't paying attention!)
- Our group spent time also talking about background color and contrast suitability online versus in-person and whether they would be different.
 - Most of the time was actually spent with introductions.
-
I liked seeing the different approaches to teaching
I really need to consolidate my screens,and just have the notes and the activity I am teaching and shut everything else down.

Keypoints:

- Live coding forces the instructor to slow down.
- Coding-along gives learners continuous practice and feedback.
- Mistakes made during participatory live coding are valuable learning opportunities.

Preparing to Teach

https://carpentries.github.io/instructor-training/18-preparation
Questions:

- How should I prepare to teach?

Objectives:

- Critically analyze a learning objective for your workshop.
- Identify checkpoints in a lesson for formative assessment.

Building Teaching Skill

Over-preparing on technical content can be tempting. Don't forget to prepare to teach!

A note on cutting: This episode is a common place for Trainers to plan cuts while preparing to teach. That’s not because this is not important – this page is a valuable resource – but we feel this is one of the sections that trainees can use effectively as a resource when actually preparing for a workshop, even without spending a lot of time doing activities on this material during their Instructor Training event.

Anticipate Your Audience

Image: A tree diagram of Carpentries instruction and audience in which Instructor Trainers teach Instructors and Instructors teach Learners https://carpentries.github.io/instructor-training/fig/instructor-training-program.png

-- -- *Exercise: Imagine a Learner* -- --
Take a moment to silently imagine a learner who might attend your workshop. What is their background? What problem do they face? What will they gain from attending your workshop?
This exercise should take about 2 minutes.

You will never know everything about the whole people who come into your classroom.
Thinking deeply about learners as people can help you prepare to bring your best self and provide an inclusive environment for everyone.

Remember Your Pre-Workshop Surveys

Examine Learning Objectives

-- -- *Exercise: Evaluate Learning Objectives* -- --
Select one learning objective from the episode you’ve used for teaching practice. Copy it into the etherpad then add numbers below your objective to address the following:

- Write your learning objective in the Etherpad.
- Suppose a learner had mastered this objective, and wanted to try something more cognitively challenging on the exact same topic (i.e. not a next step in a workflow). Identify an objective they could work towards next.
- Suppose a learner struggled to meet the specified objective. What might they be missing? Identify one more fundamental thing a learner needs to be able to do in order to be successful in meeting this objective.

This exercise should take about 10 minutes.

Beware the Urge to Complicate

-objective: Understand how SQL can be used to query databases. if they want to work further: Understand how to build queries, using SQL keywords such as DISTINCT and ORDER BY. if the original was too advanced: Explain what SQL is and why to use it, along with an understanding of relational databases.

R for Reproducible Scientific Analysis, Objective: Append two data frames

- a more cognitive challenge: append two data frames if they have a matching column value
- fundamental: understand that when appending columns between two dataframes, the dimensions must match, e.g. the number of rows in one data is the same number of rows in the second

R for Reproducible Scientific Analysis > Add and remove rows or columns

1. Add a row or column at *any particular position* rather than at "the end", Remove all columns/rows *except* a particular one
2. Bind objects to names, use functions, understand the difference between lists and (atomic) vectors, understand that a data.frame is rectangular and new rows/columns should - Objective: Indexing and slicing data. If students decide to move forward I would select integration of 2 datasets, transformate the column names and also execute some subset of data from the new dataset. Also, transformation of the type of data that each column has could be a good idea.

Starting with Data: Python -- Objective: "Access and summarize data stored in a DataFrame".

- If a student mastered this objective, they could try grouping by multiple columns and drawing a conclusion about the data with respect to these groups.
- If a student struggled with this objective, they might be having trouble with the syntax for accessing columns. They should try to access a single column of the dataframe first before moving forward.

- Objective: Version Control with Git: Understand the meaning of the --global configuration flag
    - How to identify a per-repository (not global) configuration item, make it a global setting that has a different value, and remove the previous per-repo configuration so there is not a conflict.
    - How to identify a setting is global using a) the git command line or b) intituition of settings that cannot be changed because they are or are not in a folder for a git repository.

Objective: it is necessary to break a MARK binary file to work with that MARCdata in the MarcEditor. The extension of theses easily readable MARC files are .mrk rather than binary extension or.mrc

- They should understand:
    - Different ways to read and write files,
    - How to work with individual bytes of data,
    - Converting data from one form to another. breaking it out of a binary format to a mnemonic.
- Without understanding these basics, learners might find it hard to convert binary MARC files into the easier-to-read .mrk format.

- Objective: Understanding bash command of scp and how to download a file from remote server into a local machine.
Learner should be familar at least terminal and server enviorment to meet that objective.
 R for Reproducible Scientific Analysis: mastered creating rmd file for reports, and adding coded ggplot2 figures, now can move onto other packages to support figures such as plotly to provide interactivity within the report html. If struggling they may be missing the concept of packages in R, related to functions and how things are done behind the scenes.

 Objective: Open Refine: Explain what Facets and Filters are

- Use facets and filters to work with a subset of data (you can explore your data using these techniques, write down any messiness you are seeing with facets and filters and think about what you need to do to clean up these anomalies)
- Locate controls for navigating data in OpenRefine (faceting and filtering are accessed in the columns of the graphical interface, perhaps a student needs a review of the structure of imported data and how it presents in the graphical interface or they need to find where navigation features are in Open Refine)

objective: shell - working with files and directories.
    * move on to the next lessons, because this is very introductory - pipes and loops, it introduces new syntax and a way of working, possibly thinking.
    * repetition and practice.

- Objective: Setting up the python environment for importing a pandas dataframe
* naming them into a virtual environment
* starting new and following the steps agian

- Read tabular data into Python using Pandas.

-import csv file using Pandas.
-try other libraries such as numpy

Feedback for day 3

One up one down:

Please write one thing below that you enjoyed or found useful today:
- more breakout room time
-Practising the live-coding sessions (agreed)
-The live coding was very useful. The pace of the workshop was great and it didn't feel overwhelming
-live coding practice was very useful and informative how unprepared one can be with this!
-I enjoyed the live coding, very useful to put what we learned in practice (at least attempt to)
The lessons were approachable and helped me with tangible next steps and resources I can use.-The live coding session was great!
- the practical aspects are a nice change
- I enjoyed the live coding
- Well-structured samples of good/better/best live-coding practices
- signing up for the checkout necessities was useful
-I appreciated the prompt to sign up for Welcome Session and Teaching Demos, helping us continue our journey
- I enjoyed learning more about why the carentries use live coding and tips to execute it better.

Please write one thing below that could be improved from today:
-I liked it, very informative. Maybe the same as for the previous days - to split up the 15min break in two breaks
-workshop organization was a bit technical and removed from us, but I guess still necessary to learn
- More advance notice for when in the day we will do the live coding session in the day (so I can clear

one screen/monitor and be ready to go)
-Having the live parts on the very first day (helps to get to know each other)(actually like this idea - we have come into day 3 and still havent really crossed paths with most of each other, to have a group session with live coding at the start would be a great icebreaker)
I want to piggyback on the comment above, a way to get to know other participants and network would be appreciated, idk if you have a method for linking folks together, but I have really enjoyed meeting the cohorts I have been partnered with the last few days.
-If time permits, make the live coding groups larger? From 3 people to 4 people perhaps.(agreed)
-
-i think it would be great to have the coding groups larger, possibly 5/6 to get different perspecitvies and feedback,
- more information on what to critique during live coding would be helpful, like a rubric or something
-I think I needed a couple more breaks earlier in the morning since that content was pretty dense (even just 5 minutes would have been great)
-
-more partical aspects being put into groups, otherwise it is alot of sit and listen.
-
-Maybe increase the live coding time to 5min :)
-

Prepare to Use Formative Assessments
Metacognition: learner awareness of their own process and progress can support continued effort beyond the workshop

-- -- *Exercise: Where are your Checkpoints?* -- --
Have a look at your learning objective again and identify *where* in the lesson that objective should reasonably be achieved.
This exercise should take about 5 minutes.

-- -- *Exercise: Assessment is for Everyone* -- --
How might you apply formative assessment to:
a) verify that that achievement has been met by all and
b) make learners aware of their accomplishment?
Keep in mind that formative assessment can take many forms, including multiple choice questions, other exercises, spontaneous questions and calls for sticky notes. Write some notes or thoughts about this process in the Etherpad for discussion.
This exercise and discussion should take about 10 minutes.

How Frequent?

- Formative assessments of some kind should ideally be used every 5 minutes and at least every 10-15 minutes
- "Do You Understand" is ineffective as formative assesment

Prepare to Cut

- Keep breaks on time
- Watch out for dependencies
- Leave time to wrap up your workshop
- Do not speed up
- Communicate with your team
- Communicate with your learners

Review the Instructor Notes

Review Prior Feedback

- Repetition vs Reflective Practice

Connect With Your Team

-- -- *Exercise: Minute Cards Revisited* -- --
Follow your Trainers' instructions to share feedback your event.

Please leave feedback from line 1649

Keypoints:

- To teach effectively, you have to know *who* you are teaching.
- Good learning objectives identify specific events that can be evaluated through formative assessment.
- A good exercise informs Learners and Instructors when an objective is achieved.

LUNCH (1h) / END 3rd HALF DAY

More Practice Live Coding

https://carpentries.github.io/instructor-training/20-performance
Questions:

- How did you change your teaching in response to feedback?

Objectives:

- Use feedback to improve your teaching.

-- -- *Exercise: Round Two* -- --

- Before splitting into groups, read the rubric that is given to Instructor Trainers as a suggested framework for evaluating the online teaching demonstration sessions that are part of Instructor checkout.

https://carpentries.github.io/instructor-training/demos_rubric.html . (Note: demos are not scored, so this rubric is for advisory purposes only.) What questions do you have?

- Return to your groups and repeat the previous live coding exercise, re-teaching the same content as before. This time, the presenter should incorporate changes based on feedback received, and everyone should try to 'level up' their feedback using the rubric for teaching demos.
- When you are finished, add some thoughts on this process to the Etherpad: What did you change? Did it work better or worse with the change? How might you do it if you were to teach it again?

This exercise should take about 10 minutes for rubric discussion, 25 minutes for teaching, and 10 minutes for de-brief.

Rubric questions/thoughts:
In the rubric, "Displays lesson materials/notes on their screen" is noted as opportunity for growth. Does this mean we are discouraged from pre-structuring our code using comments?

As far as I recall, it is advised to use the full given script (e.g. while teaching R lessons) to guide the students through the material. I feel the given script is heavily overloaded for beginners and I personnally would rather only give them the full script afterwards (i.e. remind them of it and take some minutes at the end of the workshop to go over a completed form) and would reduce the typing to the code to the bare minimum during the session itself. Any thoughts or recommendations here?
https://speakerdeck.com/minecr/let-them-eat-cake-first-14c0fcf0-4fe1-4e80-9c41-a7813e842538

I meant it differently... i wouldl not use the full script in R Studio all the time. As it fills up a lot of space. However, I can show them the file first, of course (and they will download it)

Speaks and types at an appropriate (slow enough) pace, it is a good reminder that perhaps is good to maintain a steady pace when we present a material.

Mirrors the learner's environment - is there an option in RStudio to reset to the default or what is the default orientation to make note of this in preparation for future workshops
The default for RStudio is also really small when screensharing/presenting
When I am teaching R, I usually show the audience at the very beginning how I changed the appearance in the Global Options + think about the language!

I thibnk you can reset to the defaults Rstudio: https://support.posit.co/hc/en-us/articles/200534577-Resetting-RStudio-Desktop-s-State

- - this looks more like the directory of where RStudio is stored and its data rather than the appearance of the workspace which is my driving question

What did you change? Did it work better or worse with the change? How might you do it if you were to teach it again?

-I ironed out the Jupyter notebook vs. terminal issues, and presented the same environment the students would use. Spent less time waxing poetic about Python, and gave succint info on setting up their project folder. This went much better than yesterday, and I had more time to discuss actual coding. I like the pacing, and would keep this change going forward if I taught it again.
- I cut down some of the introduction that I had before and just briefly explained what we were about to do before getting to the live coding.
-I made my text larger and made sure to welcome the group/ recap what was learned in the last lesson and how that would play into the objectives for that episode. I would tweak my terminal set up for the next session.
-Jumped straight into the coding as taken the intro out to that episode.
Changed the background colour and increased the font size
-I included an intro how one might customize their RStudio layout since it would be different than the default. It worked but I think I spent too much time explaining it since I was only really able to get one line of code written before my time was up. So in the future before workshops I'll make sure I have the default set up.
-
-I focused on staying in just one window instead of switching between the lesson script and the coding environment, which helped focusing me and my lesson.
- I am trying to do an error intentionally that mostly common in my topic to show how to handle it.
-I had the notes printed which was a game changer for me - I was much more focussed and concentrating more on the typing + it helped that another participant already introduced the data and I could directly jump into teaching
My first training session was a challenge, I attempted to jump into a lesson and did not refresh the MarcEdit tool. I received good feedback to recap the previous lesson for the audience.I tried to keep light of the snafu, and explain that refreshing or restarting a tool before use is a good practice ;)

I cut down some of the explanation around messy data and was able to cover more ground in the lesson.

It was much better this time, I got straight to the point. However, my pace was fast and I usually combat this but frequently asking my students if my pace is fine.
How important is it to stick to the script?
Questions:

- Emily: When teaching importing data to Python, what will be your recommendation—downloading data locally for live code, or directly importing from online link? I have got two different feedback in these two rounds

  - - Rob: This is usually covered in the episode/lesson setup. In most cases we would expect learners to have downloaded everything they need before the workshop. Directly importing from online links can work fine as long as you have a good internet connection. Some lessons require that you load in data locally as it's also teaching learners about file paths etc

For our demos, should we include time for the learning objectives of an episode as that takes some time away from the 3 mins today, so i removed that and just got straight to the live coding. Thanks

- - Rob: The demos are 5 mins, so you can take 30 seconds to introduce what you will be live coding, but I wouldn't take any more than that.

For the demo I am considering (R for Reproducible Scientific Analysis: Producing Reports With knitr (swcarpentry.github.io)) following the File New R Markdown in RStudio gives a outline document that then should be ignored/deleted, to move onto the coding, could it be ok just to create a blank R doc and save it .rmd instead to avoid that happening. Doesnt seem a positive thing to create a template doc in front of learner to then just delete or is that ok? great thanks!

If the material and screensize permit it, re the rubric, should we have the browser with the episode and a coding window side-by-side? The rubric talks about showing notes, but I assume that is kind of different.
OK re the rubric what does "notes" mean as areas for growth in the rubric?
Keypoints:

- (Reflective) Practice makes perfect.

Working With Your Team

https://carpentries.github.io/instructor-training/21-management
Questions:

- What are the challenges of managing a heterogeneous classroom?
- What should we do if there is a Code of Conduct violation?
- What does it mean to be a co-Instructor?
- How does an instructional team prepare for a workshop?

Objectives:

- Identify potential challenges of teaching learners with very different backgrounds and skill levels.
- Locate resources to direct your response if someone at your workshop violates the Code of Conduct.
- Identify workshop roles and responsibilities for your team.
- Use The Carpentries workshop website template instructions to start creating a website.

Never Teach Alone

The Instructional Team

- A *Host* who organizes the workshop logistics
- Two or more *Instructors* who plan and execute workshop instruction
- *Helpers* who support learners during the workshop

Hosting
https://docs.carpentries.org/topic_folders/hosts_instructors/hosts_instructors_checklist.html#host-checklist

Helpers

- help learners with setup and installation
- answer questions during exercises
- monitor the room to spot people who may need help (indicated by a sticky note or otherwise)
- monitor the shared notes and either answer questions there or remind the Instructor to do so during breaks

https://docs.carpentries.org/topic_folders/hosts_instructors/hosts_instructors_checklist.html#helper-checklist

Carpentries Classroom Practices

- Starting with the Code of Conduct
- Participatory Instruction & Hands-off Help
    - Learners Use Their Own Machines
- Sticky Notes
    - Accessibility of Sticky Notes
- Formative Assessment
- Breaks (ideally with snacks)
- Feedback

Co-Teaching Models

- Team teaching: Both teachers deliver a single stream of content in tandem, taking turns the way that musicians taking solos would.
- Teach and assist: Teacher A teaches while Teacher B moves around the classroom to help learners.

Sticky Situations 1: Learners at Many Levels

-- -- *Exercise: What Are the Challenges?* -- --
What are some of the challenges you might expect when teaching learners with a broad range of expertise? Add your thoughts in the Etherpad.

 -
 -Boredom and anxiety, not good feelings for learning. I think it would be good to have extra exercises for advanced learners, something that helps preview upcoming lessons. That way you stick with the learners who really need help, but engage the advanced students in something that helps you in the future.
 -Some learners will feel bored while others will feel overwhelmed. Finding the right balance is difficult.
 - Confident students who will want to insult technology ("I hate Windows! macOS is the best!") and will not stop even if the instructor explains that is discouraged.(agreed and actually heard that once in a workshop)
 -advanced learners who help others in a way they don't want (overreaching, taking the keyboard, unwanted explanations)
 learners with disabilities may not always be obvious, we had a deaf student on a learning course who was

fantastic at lip reading, but needed to see you talk
-People who are not really participating, coming/going whenever they want, then having questions when parts are actually already over...
When more knowledgeable learners start doing something else and tune out of the lesson, this could distract a novice learner.
those who are not as advanced as others in the room may feel intimidated
-Frustration from more advanced students and having to fight the urge to speed up to accomodate those who feel you are moving too slow.
- Have to be cognisant of the pace, and when asking for pacing feedback from learners, make sure to hear from those who are less experienced.
- Problems with the (teaching) language (either real problems or they just think 'I am not good at English...')
- Giving a workshop for a group of students where they are told they have to participate and have a wide range of skill levels. how do you keep the advanced people occupied while also not going to fast for the noivces (have limited staff).
- People going ahead in the material and then getting stuck and asking questions.
- Discomfort installing, not just using, new software.
- I live with a teacher and she says a frequent challenge is fundamental technical literacy for the younger generation despite being Internet native: what is a file? You don't just leave "everything" in a Downloads folder, you move it? (This comes up daily for high schoolers, which is surprising to me.)

- - Rob: This is a *really* good point and something that there have been discussions about in the community. There is an (empty) Incubator lesson that would be the place to start this, but there hasn't been any progress as yet: https://github.com/carpentries-incubator/foundational-computer-skills

This discussion should take about 5 minutes.

- workshop advertising
- exercises
- partnering
- managing the conversation
- helper vigilance

Sticky Situations 2: Code of Conduct Violations

A critical function of the Code of Conduct is to ensure that our community does not tolerate or encourage the persistence of harmful behaviors. In order for the code to work well, incidents must be reported. Note that it is not the responsibility of the reporter to determine whether a Code of Conduct violation has occurred; when in doubt, it is best to report an incident and allow the Code of Conduct Committee to make that determination.

-- -- *Exercise: Know Your Resources* -- --
1) Take 5 minutes to read through the Code of Conduct Incident Response Guidelines: https://docs.carpentries.org/topic_folders/policies/incident-response.html

2) Discuss what you have read in small groups. As questions arise, you may wish to refer to our complete Code of Conduct section in The Carpentries Handbook: https://docs.carpentries.org/topic_folders/policies/index_coc.html or to the Transparency Reports released by The Carpentries Code of Conduct Committee: https://github.com/carpentries/executive-council-info/tree/master/code-of-conduct-transparency-reports

- What kinds of things could your instructional team agree upon in advance of your workshop?
- What questions do you have about CoC enforcement?

3) Write some notes in the Etherpad.

What happens to the bad actors in terms of repercussions?

- https://docs.carpentries.org/topic_folders/policies/termed-suspension.html

How often do the committee receive incident reports, are there more reports of online instances than in person

- - The reports are received by the committee usually within a week, but if it is a serious violation the host or instructor/organiser may/should take immediate action.

This discussion should take about 10 minutes.

Know your Local Laws and Policies

Planning Together

-- -- *Exercise: Teaching Together - Nuts and Bolts* -- --
With a partner, imagine that you are planning a workshop together. For this exercise, you may assume that your workshop has a separate, designated Host.

- How would you prepare to teach a workshop together?
- How would you coordinate with other members of your instructional team (e.g. Host, Helpers)?
- What kinds of things will you do to support each other during the workshop? What won’t you do?

Record some notes, and share your thoughts with the group. This exercise should take about 10 minutes.

Setting up a Workshop Website

-- -- *Exercise: Practice With The Carpentries Infrastructure* -- --
For this activity, your Trainer will put you in groups, but you may choose whether to work together or independently. If you work independently, you can still use your group as a resource to ask questions as they emerge.

Go to the workshop template repository: https://github.com/carpentries/workshop-template

- If you have a GitHub account (or don't mind creating one) and are comfortable doing so, follow the directions to begin creating a workshop website using your local location and today's date.
- Alternatively, have a look at the video tutorial linked on the instructions page. With any time remaining, check out the websites for upcoming Carpentries workshops on our website: https://carpentries.org/upcoming_workshops/
- Add your questions and thoughts on this process to the Etherpad. If you created a workshop website, add the link there as well.

This exercise should take about 15 minutes.
- https://benkeene.github.io/2024-03-29-florida-online

- https://lauraschild.github.io/2024-03-29-test-online/
- https://frm1789.github.io/2024-06-01-r-ladies-chicago-online-session/

-
- https://angeliamiller.github.io/2024-03-29-trial-online/
Note: Sometimes web browsers will cache the workshop webpage, so when you make changes in GitHub, they do not show up on the workshop webpage immediately. Two ways to avoid this are to use a "private" or "incognito" mode in your web browser or by following these instructions to bypass your browser cache: https://en.wikipedia.org/wiki/Wikipedia:Bypass_your_cache

Setting up an Etherpad

Keypoints:

- Team work takes work, but allows you to share the load and build connections.
- Working with a broad range of learners can be challenging, but there are many ways to keep a classroom happy and motivated.
- The instructional team decides how to respond to Code-of-Conduct incidents during a workshop; all violations should be reported to The Carpentries Code of Conduct committee for follow-up.

BREAK (15 min)

Launches and Landings

https://carpentries.github.io/instructor-training/23-introductions
Questions:

- How do you actually start a workshop?

Objectives:

- Connect goals of an introduction with options for content and delivery.
- Practice a short introduction.
- Identify worthwhile elements of a workshop conclusion.

Launching your Workshop: The Introduction

"primacy effect": a tendency to remember things presented at the beginning of a list or event

-- -- *Exercise: What is in an Introduction?* -- --
Get into small groups (3-4 people) and discuss the questions below. Take notes on your answers in the Etherpad.

- **What do you hope to accomplish in a workshop introduction?**
- coc
- carpentry values: open space and comfortable learning environment
- schedule
- content overview
- feedback opportunities
- check software requirements/remind students of necessary software
- mention pre-workshop survey
- introduction of students, icebreaker
- introduce the instructors
- let the learners get to know each other so they find it easier to ask each other for help later
- let the participants introduce themselves
- Tell them what you're going to tell them, clear outline of what you'll be covering, why it's relevant, hype it up!
- codes of conduct
- materials / sources
- channels for asking questions
- introduce the sticky note system
- introducing instructors, giving a general idea of the content to be covered, when breaks are scheduled, introducing the helpers and their role, explaining how the workshop will be delivered, code of conduct
- introduce the venue (where are the toilets, where can they get food/water)
- what do people need to be able to attend successfully (assistive technology etc., though that should best be surveyed before the workshop starts)
- Give an overview of the objectives of the workshop
- Mostly hybrid workshops here so we need to introduce helpers online and in person.
- **What information do you need to include in an introduction to accomplish these goals?**

- audience: what's their background
- offer additional materials
- plan of the venue (if workshop is in person)
- schedule
- teaching format
    - i.e. lecture vs group work timing
- expectations of students (participation)
- desired workshop outcomes
- allll the links (etherpad, website, ...)
- logistical information
    - food, restrooms, breaks, etc.
- Brief summary of content, pacing, what the learners can expect to discover

After 5 minutes, come together, and combine ideas as a large group.
Finally, compare your ideas with the list of topics below. Did you miss anything? Did you come up with something that is not listed below?

Learning Objectives For your Introduction

After the introduction learners should:

- be able to predict the type of instruction
- know what will be taught
- understand what will be required of them
- believe that they can learn from the workshop

The instructional team should:

- know of who is participating in the workshop and what their expectations are
- have an initial impression of how learners respond to participation prompts and what will be needed to encourage them to engage

Setting the Stage

- attire
- physical environment
- time before class
- introducing yourself
    - introductions for everyone
- doubts
- seeding a classroom community
    - icebreakers

Teaching Your Trajectory: Workshop 101

- Describe the prerequisites (if any).
- Share the schedule and logistics
- Communicate the workshop structure
- Communicate your expectations for learners, including:
    - how to follow the Code of Conduct
    - ways to ask for help
    - ways to give feedback to the instructional team

- Collect and share baseline data on learners
- Share some advice for success
- Whet learners' appetites for workshop content

-- -- *Exercise: Practice Your Introduction* -- --
Imagine you have completed instructor training and you are about to teach a full lesson around the material you have been practicing teaching today.

- Write out some notes, covering a few of the topics described above:
    - Introduce yourself effectively
    - Clarify learning objectives and expectations
    - Set the tone for the workshop
- Return to your groups of 2 or 3 and each give 2 minutes of your introduction. (5-6 min)
- After each introduction, briefly share feedback, reserving extensive discussion for after all have had a turn to present.

How did you find delivering your introduction? DId you get any feedback?
-
-
-stressful :D but good practice
-
-
-
-
-
-
-

This exercise will take about 15 minutes.

The Art of a Smooth Landing

-- -- *Exercise: Brainstorm: Making the Last Moments Count* -- --
You have made it to the end of your workshop! Everyone is exhausted and their brains are full. You could cover more content… or you could use the last few minutes in another way.

In the Etherpad, write down one thing you could do at the end of a workshop. What is the value of spending time on that thing? If you have time after writing down your idea, read through the others in the etherpad. If you have another idea that has not been written down yet, add it to the list.

This exercise will take about 5 minutes.

-

- Remind to fill out the post workshop survey
- Ask learners to pause and reflect on what they've learned -- ask them to think to themselves 1-2 things they might apply in their work/research/studies. Moves from the actively learning portion of the workshop to a 'consolidation of knowledge' step. Then move on to formalities and closing comments.
- summarize what was learned and the goals that were achieved
- if we missed some content: let them know how to catch up on it
- further resources: this is just the start of a learning journey, remind students that you always learn new thing when programming and that googling and looking things up is normal and good
- provide opportunity for feedback then and through a post-workshop survey
- Let people know where they can get help or further support (if you have those resources within your organization)
- Recap of the contents, reminder to fill the workshop survey.
- Recap; offer resources for further skill development, let students know the policy of sharing the workshop materials
- Like the above comment, maybe resources on how to debug, more generally how to confront roadblocks and empower themselves to try solutions and get help (not just about course material)
- let them write a card for themselves with a short note on a) the main content they learned in the course and b) the next step they want to to (with respect to course content) to build on it

thank learners for attending, how to move forward with feedback, reminder where they can find the lessons/code and forthcoming Carpentries events that may interest them, or point out where to find more info such as the Carpentries site and local information on training. Thank the co instructor(s) and helper(s). Ensure they dont leave laptops etc behind.

- How to go from here: possible next courses, possibilities to stay in touch
- Being an instructor or teachers always seems scary to me but after having that traning now i feel i can manage the all it from many aspects. So that my aim to develop my teaching skills with Carpentries cummunity and getting involve more and more !
- Show resources (both locally and online) to continue learning or get a refresher. I would also plug my own services (library) and other carpentries workshops. I would also share the post workshop survey
- It would be interesting to take some time at the end of the lesson to think about how your future self could apply the lessons learned tomorrow in your work or personal life.
- Wrap up and proviide takeaways. Ask for feedback. Provide suggestions for future learning.
- Ask students what they are going to take away and put into practice or you could also ask what inspiration participants have found during the workshop, that may be helpful for some of the lessons and not for all
- Ask learners to identify something they improved or learned from the workshop, something/a concept that they grasped really quickly or do really well, and something that they're going to do with their new skills after the workshop (self effort/improvment/performance feedback)
- Thank everyone that helped out with the training like helpers, co-presenter, host etc.
- Maybe plug relevant further training
- 'house keeping' - to take with them any leftover food :-) (in person workshops)

Keypoints:

- A planned introduction is key to creating a functional workshop environment.
- Conclusions support reflective practice and set the stage for continued learning.

Putting It Together

https://carpentries.github.io/instructor-training/24-practices
Questions:

- How are the teaching practices we have learned used in our workshops?

Objectives:

- Organize your knowledge of teaching practices and create a plan for using these practices in a Carpentries workshop.

-- -- *Exercise: Picking up the Pieces* -- --
Based on the content we've discussed throughout this workshop, add at least one item to each category below:

- Concepts/Theories
- Tools/Practices

This exercise can be done as a class and should take about 5 minutes.

-- -- *Exercise: Organize Your Knowledge* -- --
Use a concept map or other visual organiser of your choice to connect some of the concepts above. You don't have to use them all! How are the terms you have chosen to include related to each other?
Work on this on your own. There is no "right answer" – this is about you building up a mental model, moving from "novice" to "competent practitioner".
If you feel you have finished organizing your thoughts, try the next exercise.
This exercise should take about 5 minutes.

-- -- *Exercise: Parting Thoughts (optional)* -- --
If you did not think about these issues when organizing your topics in the previous exercise, now consider:

- How would you describe your mental model of teaching?
- Can you identify why each topic above applies to teaching for the Carpentries?

Keypoints:

- Having a plan makes it easier for you to remember to implement the important teaching practices you have learned.

Wrapping Up

https://carpentries.github.io/instructor-training/25-wrap-up
Questions:

- What can we improve in this training?

Objectives:

- Reflect on the course.
- Articulate constructive feedback.

-- -- *Exercise: One Up, One Down* -- --
Provide one up, one down feedback on the entire Instructor Training course. Remember:

- Say only one thing, and try not to duplicate. This gets harder for those who come later!
- Trainers should try not to respond, only record responses (e.g. in the Etherpad). This is also hard, but important!

This exercise should take about 10 minutes.

One thing you enjoyed from the whole course:
Enjoyed the sessions with interactions
-Breakout rooms, getting to meet people.
-Rob and Anne were great! Thank you!
-
-the possibility to exchange comments with others and receive feedback. Also, I love to have a template and definitely I am planning to use it.
- Loved communicating with other attendees, very collaborative and supportive. The instructors did a great job outlining the process of joining and contributing to the Carpentries.
-having to demo teaching was very helpful to run into some first hurdles, fix them and feel more comfortable about teaching now
-discussion and learning from other students
-Enjoyed and learned a lot all course and really feel confident to take a step to real-teaching world.

One thing that could be improved:
-More code sessions. Maybe a redo of the first code session, because after the feedback I would like to try again the same session but adding the positive contributions that other participants made.
-More interaction with other people in the training
- Less interaction with random sets of people, found it extremely challenging to adapt to different "breakout-room-sets" and/or complete introduction of everybody at the very beginning of the workshop
-Longer teaching demos to get more useful feedback
-Having more episoed and live coding demos
-, overall I would suggest being able to have an icebreaker for everyone, we havent met each other through all the breakout rooms, but it would have been great to get everyone introduced for a couple of minutes and the motivation for wanting to be am instructor.We have spent 4 afternoons
together :)seconded
-the etherpad is hard to navigate and not good for my brain
-

-- -- *Exercise: Minute Cards (optional alternative)* -- --

Please use your minute cards (sticky notes or virtual) to give your Trainers anonymous feedback directly. This exercise should take 5 minutes.

-- -- *Exercise: Post Workshop Surveys* -- --
Assessment is very important to us! Please take the remaining time to complete this ~5 minute post-workshop survey.

Thank You!

Keypoints:

- Feedback applies to all kinds of learning, including learning how to teach.

Before You Leave

Please fill out the post-training survey at https://carpentries.typeform.com/to/cjJ9UP#slug=2024-03-26-ttt-online-CET