Users are expected to follow our code of conduct:
https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

Instructors:

- John Simpson, University of Alberta / WestGrid / Compute Canada, john.simpson@computecanada.ca, @symulation
- Naupaka Zimmerman, University of San Francisco, nzimmerman@usfca.edu, @naupakaz

Observer:

- Guilherme Castelao (Gui), UC San Diego, castelao@ucsd.edu
- Reed Cartwright, ASU, cartwright@asu.edu, @MinionLab

Sign in: Name, Institution, Email, Twitter (optional)
Please sign in so we can record your attendance.

- Katie Coburn, UC Merced, kcoburn@ucmerced.edu, @coburn_m
- Mara Sedlins, Colorado State University, Mara.Sedlins@colostate.edu
- Jacob Deppen, University of Washington, jdeppen@uw.edu @jacob_deppen
- Adriana Signorini, UC Merced, asignorini@ucmerced.edu
- Elizabeth Salmon, UC Merced, esalmon@ucmerced.edu
- Matthias Bussonnier, UC Merced, mbussonnier@ucmerced.edu, @mbussonn
- Jeffrey Weekley, UC Merced, jdweekley@ucmerced.edu @jdweekley
- Sarvani Chadalapaka, UC Merced, schadalapaka@ucmerced.edu, @schadalapaka
- Chris Olson, UCSD SIO, cjolson@ucsd.edu
- Martino Sorbaro, University of Edinburgh, martopix@gmail.com, @martopix
- André Palóczy, UC San Diego, apaloczy@ucsd.edu
- Anne Glaser, Bayer Crop Science, anne.glaser@bayer.edu
- Billy Broderick, New York University, billbrod@nyu.edu
- Dima Lituiev, UC San Francisco, d.lituiev@gmail.com, @DimaLituiev
- Craig Boman, Miami University (Ohio) bomanca@miamioh.edu, @craigboman
- Juliana Velez Lujan (UCSD) jvelezlujan@ucsd.edu
- Julia Piaskowski, jpiaskowski@uidaho.edu, University of Idaho
- Nohemi Huanca-Nunez, University of Nebraska, nohemihuanca@gmail.com, @nohemihuanca
- Saranya Canchi, UCSD, scanchi@ucsd.edu, @S_Canchi
- Eva Sanchez, UCSD, e9sanche@ucsd.edu

I. Welcome

Code of Conduct: https://software-carpentry.org/conduct/

Introductions

Information for Today’s Learners

1. Add your name to the Google Doc (above)
1. Introduce yourselves! In your introduction, (a) explain your work in 3 words and (b) say something you are proud of (not necessarily related to research or teaching).

Our First Exercise (2 min)

In the Google Doc, write down your name, the best class you ever took (or one class from your top ten, if you can't decide), and what made it so great.

- Katie Coburn: introduction to graduate-level statistics; good connection with teacher, clear concepts
- Mara Sedlins: AP Physics in high school -- lots of memorable in-class demonstrations, hands-on assignments
- Jacob Deppen - Zooarchaeology - oriented around hands-on practice with real animal bones from archaeological sites that was the best approximation for what it is like to do zooarchaeology in "the real world"
- Adriana Signorini - intro to stats- why? I needed it to do my job -
- Elizabeth Salmon: Public Libraries- peer to peer learning, each class session followed a similar structure
- Matthias Bussonnier: Handwaving Physics – Teaching Physics only with Order of magnitude / rules of thumbs/ Dimensionality analysis.
- Jeffrey Weekley - Introduction to Networking (graduate level course). It was very early days of TCP/IP, so we learned how to write data packets, decode them, engineer network features. It gave me a solid foundation for my research thereafter.
- Sarvani Chadalapaka - Embedded Systems because it was VERY hands-on, technically demanding but fun class!!
- Chris Olson - Control Systems - This had both theory and hands on portions. I found the topic very interesting and the teacher managed to explain it in a way that just clicked for me.
- Martino Sorbaro - (Symbolic) Artificial intelligence: good "acting" from the lecturer, formal rigour
- André Palóczy - Geophysical Fluid dynamics during the MS. Instructor was outstandingly clear and organized. Conveyed physical concepts very well.
- Anne Glaser: Intro to Physics, the instructor gave many real world examples that made it easy and interesting to learn
- Billy Broderick: Intro to Combinatorics, because it was lots of problem solving, requiring us to think about problems / how to use what we learned from a variety of angles
- Dima Lituiev -- Spatiotemporal Modeling and Simulation (Ivo Sbalzarini, ETHZ, CH). Why: both skill and knowledge oriented, good balance of theory and practice, steep gain of knowledge but very well guided.
- Craig Boman - Philosophy of logic, because it taught me how to think logically
- Juliana Velez Lujan: Clinical Correlation, because it connected several systems of the human being to diagnose diseases.
- Julia Piaskowski: applied linear models - got me very excited abouts statistics, new way of looking at linear models
- Nohemi Huanca-Nunez- Introduction to Ecology because it was very clear in the goals and expectation of each class
- Saranya Canchi - Functional Neuroanatomy as it inspired me to focus on neuroscience for my research i
- Eva Sanchez. Biology. That day I decide to pursue a career in Biology

A Brief Overview of the Carpentries

Instructor Training Workshop Overview

- How learning works
- Building teaching skill
- Creating a positive learning environment
- The Carpentries: history and culture

Assessing Trainee Motivation and Prior Knowledge

Background

Have you ever participated in a Software Carpentry or Data Carpentry Workshop?

- Yes, I have taken a workshop.
    - Anne Glaser
    - Matthias Bussonnier
    - Juliana Velez
    - André Palóczy
    - Jacob Deppen
    - Saranya Canchi
    - Elizabeth Salmon
    - Eva Sanchez
- Yes, I have been a workshop helper.
    - Katie Coburn
    - Dima Lituiev
    - André Palóczy
    - Matthias Bussonnier
    - Elizabeth Salmon
    - Chris Olson
    - Saranya Canchi
    - Mara Sedlins
    - Jeffrey Weekley (twas an NSF workshop)
    - Eva Sanchez
- Yes, I organized a workshop.
    - Katie Coburn
- No, but I am familiar with what is taught at a workshop.
    - Billy Broderick
    - Julia Piaskowski
    - Sarvani Chadalapaka
    - Martino Sorbaro
    - Craig Boman
    - Nohemi Huanca-Nunez
    - Chris Olson
- No, and I am not familiar with what is taught at a workshop.
    - Adriana signorini

Which of these most accurately describes your teaching experience?

- I have been a graduate or undergraduate teaching assistant for a university/college course.
    - Katie Coburn
    - Billy Broderick
    - Jacob Deppen
    - Dima Lituiev
    - Martino Sorbaro
    - André Palóczy
    - Matthias Bussonnier
    - Mara Sedlins
    - Saranya Canchi
    - Nohemi Huanca-Nunez
    - Jeffrey Weekley
    - Eva Sanchez

- I have not had any teaching experience in the past.
    - Juliana Velez. I have just trained students in the lab but not formal teaching.
- I have taught a seminar, workshop, or other short or informal course
    - Jacob Deppen
    - Katie Coburn
    - Elizabeth Salmon
    - Julia Piaskowski
    - Saranya Canchi
    - Nohemi Huanca-Nunez
    - Sarvani Chadalapaka
    - Matthias Bussonnier
    - Martino Sorbaro
    - Dima Lituiev
    - Mara Sedlins
    - Jeffrey Weekley
    - Eva Sanchez
- I have been the instructor-of-record for my own university/college course.
    - Katie Coburn
    - Julia Piaskowski
    - Jeffrey Weekley
    - Eva Sanchez
- I have taught at the primary education level.
- I have taught informally through outreach programs, hackathons, laboratory demonstrations, and similar activities.
    - Katie Coburn
    - Jacob Deppen
    - Anne Glaser
    - Elizabeth Salmon
    - Matthias Bussonnier
    - Craig Boman
    - Dima Lituiev
    - Saranya Canchi
    - Jeffrey Weekley
    - Sarvani Chadalapaka
    - Chris Olson
    - Eva Sanchez

Which of these questions assesses flaws in a student’s mental model of a domain? You don’t need to provide answers for these questions.

- I’m not sure what a mental model is.
    - Julia Piaskowski
    - Chris Olson
    - André Palóczy
    - Craig Boman
    - Eva Sanchez
- “In Python, what is the expected output for the following statement: 1 + ‘2’” (a) ‘12’ (b) TypeError (c) ‘3’ (d) 3
    - Mara Sedlins
    - Matthias Bussonnier
    - Martino Sorbaro
    - Dima Lituiev

    - Katie Coburn
    -
    - Billy Broderick
- “Rate your experience with the R programming language.” (a) never used it (b) beginner (c) intermediate (d) expert
    - Jeffrey Weekley (a)
- “What does the Unix command ‘cut’ do?” (a) Extracts sections from each line of input. (b) Sorts fields of a line (c) Searches the input file for lines containing a match to a pattern (d) Removes a given input from a line
    -
    -
    -
    -

Key Points:

- Software and Data Carpentry are communities 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 general teaching pedagogy and how it applies specifically to Software and Data Carpentry.
- Learner motivation and prior knowledge vary widely, but can be assessed with a quick multiple choice question.

II. How Learning Works: The Importance of Practice

The Carpentries Pedagogical Model

Acquisition of Skill

- Novice
- Competent practitioner
- Expert

Mental Models

In the Google Doc, write your primary research domain or area of expertise and some aspects of the mental model you use to frame and understand your work. What concepts/facts are included? What types of relationships are included?

Your Mental Models

- Katie Coburn: meta-analysis/publication bias -- effect sizes, distributions, variability, survival, probabilities
- Mara Sedlins: data curation and archiving -- quality of data documentation, types of data that can be shared publicly, discipline-specific metadata schemas, preservation considerations for different file types
- Jacob Deppen - basic geological principles; different models of culture and behavior; taphonomy (how the archaeological record is formed); “how do we know if something is old?”; how archaeological fieldwork methods impacts archaeological finds; abstracting complex processes like “culture” as computer code
- Adriana Signorini - assessment cycle - steps in the process-

- Elizabeth Salmon - strategic searching (concept maps, filters, boolean, experimentation), information literacy (framework), source evaluation(contextual authority)
- Matthias Bussonnier
  - How Jupyter notebook works, in particular “notebook(s)”. What the difference and relations are between the web-application, notebook server, and kernel and file document. Most of the time the 4 of those are referred to as “notebook”.
- Jeffrey Weekley: Image acquisition, processing and display: analog-digital boundary (quantization and Fourier transformations), color theory, data structures, networking, information theory, physics of light and optics.
- Sarvani Chadalapaka
  - How does this code work and how can we make it more efficient (best practices?)
  - How large is the data and how are they storing/backing it up
  - What is the big picture? Connecting the dots (what we code now vs what to accomplish)
  - Documentation practices
- Chris Olson - Oceanographic data management - data/metadata structures, storage methods, archival, documentation, web services, sensor calibration, data validation
- Martino Sorbaro - Computational neuroscience - Statistics of neural activity (physical and computational aspects)
- André Palóczy - Physical oceanography, regional characteristics of a system (specific, measured data in space [e.g. shipboard data] and time [time series, e.g. moorings]) and the dynamical laws that describe their behavior (general).
- Anne Glaser: data science platform, how the various products are used together and how information/documentation is shared
- Billy Broderick: mental model largely consists of different components in the visual system and the subset of information / computations that they carry out. E.g., eye/retina as a whole absorbs light and does some pre-processing, including converting luminance (physical measurement of number of photons) to contrast (relative measure of difference in luminance across image). Important computations done by different portions: linear filtering, rectification, normalization (so these are connected to different anatomical concepts, I think).
- Dima Lituiev: machine learning models, model training, generalization, underfitting and overfitting, parameter optimization, data structures (tables, text, raster images, vector graphics), data modalities (genomics, histology, radiological studies, pathology notes, radiology notes).
- Craig Boman - Functional Requirements for bibliographic description (FRBR) - how do we describe physical/digital resources based on how it is expressed.
- Juliana Velez Lujan. Leukemia. Genetics, microenvironment, differential biomarkers. How these and other factors are connected in the development and maintenance of leukemia.
- Julia Piaskowski: agricultural researcher, understanding study objectives and identification of appropriate statistical model(s) to meet study objectives
- Nohemi Huanca-Nunez: Forest regeneration, plant species interactions
- Saranya Canchi - Alzheimer’s research, statistical modeling, transcriptome analysis to understand genotype- phenotype correlation and identify population subtypes
- Eva Sanchez. Metabolism of green microalgae. Genomics and transcriptomics. Reconstructing/predicting metabolic pathways in different cells comparments. Heatmaps.

Types of Misconceptions

- Factual errors
- Broken models
- Fundamental beliefs

Identifying and Correcting Misconceptions
Formative assessment
Summative assessment

Reflective practice

Identify the Misconceptions (3 min)
Choose one wrong answer and write in the Google Doc what the misconception is associated with that wrong answer.
Q: what is 27 + 15 ?I am
a) 42
- Misunderstanding that these are 6 bits signed integer, it overflow at 32 :-)

b) 32 - Forgot to carry the remaining 1
c) 312 - (javascript) - Dropped the carried 1 in a random place

d) 33

The student tried to carry the 1 but added it to the wrong place (2 + 1) = 3, (7 + 5) = 12 = 1 + 2 = 3.

Handling Outcomes (3 min)
Formative assessments allow us as instructors to adapt our instruction to our audience. What should we do as instructors if the class votes for:
1. mostly the same one of the wrong answers?
   - Focus on the misconception that that particular wrong answer captures and clarify it for the class.
   - The instructor probably failed to explain something clearly
2. mostly the right answer?
   - Proceed (with caution, as always)
   - Break out to verify whether everyone *got the right answer through the correct reasoning*, as opposed to a flawed mental model that happened to yield the right answer.
   - Discuss why the wrong answers are wrong, to reinforce not the existing links,but concepts that are not linked.
   - Short break-out discussions to discuss right answer to help lagging students see right answer (maybe alternative points of view will help them)
   - Peer instruction
3. an even spread among options?
   - Explain again, in a different way (unlike in case 1, here we can only guess *which* other way will work better)
   - Students are probably guessing, maybe you lost all of them
   - Students debate

Modeling Novice Mental Models (10-15 min)
Create a multiple choice question related to a topic you intend to teach.
Type the question and the a,b,c,d, options into the Google Doc
Explain the diagnostic power of each its distractors, i.e., what misconception is each distractor meant to identify?

1. Katie Coburn
   - Assume you have created a data frame called “cats” in R. You want to remove the fourth column. Which of the following commands would work?
     i. cats[,-4] (This is the correct answer)
     ii. cats[-4,] (Misconception; syntax is row,column)

        iii. cats[,4] (Misconception; minus sign removes column)
        iv. cats[4] (Misconception; need row or column identifier to work)

2. Mara Sedlins
    - What type of data may NOT be shared in the CSU Digital Repository?
        i. data that contains potentially identifiable information, such as interview recordings or transcripts (correct answer)
        ii. data associated with a student dissertation (misconception: student research data is not accepted or valued)
        iii. data that involves collaborators outside the university (misconception: all contributors to a dataset must be affiliated with CSU)
        iv. data that is not in a tabular format (misconception: only tabular data is accepted)
3. Jacob Deppen
    Which of these will give you all of the different species names in the 'species' column?
    - A. surveys_df['species'].nunique() [uses a valid command that includes the term 'unique']
    - B. surveys_df['species'].count() [uses the right column, but count gives a number, not names]
    - C. surveys_df['species'].describe() [a student who isn't fully understanding might guess that a 'describe' method would describe the different values]
    - D. surveys_df['species'].unique() [correct]
    - E. surveys_df['species'] [a student who doesn't understand that they need to call a method might select this answer]
4. Adriana Signorini
    - Which is a course learning outcome?
        i. A- Identify the manifestations of power (right one)
        ii. B- Identify and implement the manifestations of power (two in one, difficult to assess)
        iii. C- Lecture on the manifestations of power (teacher's goals)
        iv. D- Manifestations of power (content)
5. Elizabeth Salmon: If you are looking for information about fires in California, what search correctly combines search strategies?
    - a) wildfire* OR forest fire* OR California - incorrect use of OR, use AND to require the CA concept
    - b) wildfires AND forest fires AND California- incorrect use of AND; combine related terms using OR
    - c) ("wildfires" OR "forest fires") AND California- incorrect use of "; use " around 2 or more words you to search for as a phrase in that exact order
    - d) (wildfire OR "forest fire") AND California- correct
6. Matthias Bussonnier
    In a unix terminal, how can I usually get back to my HOME directory if I'm lost ?
    - a) cd HOME -- exanding HOME need a $ `cd $HOME` would be good.
    - b) cd . -- . is current working directory.
    - c) cd -- correct the default argument of cd is $HOME.
    - d) cd .. -- Confusion with .. which is parent directory. Works sometime when you are only one folder deep.
    -
7. Jeffrey Weekley
    How many colors can be represented in 10-bit RGB color encoding system?

    a. 3072 (summed 2^10 for each channel)
    b. 1,000,000 (multiplied 10^2 for each channel)
    c. 1,073,741,824 (correct)
    d. 1,099,511,627,776 (included an extra channel [RGBA] e.g. 1024^4 versus 1024^3)

8. Sarvani Chadalapaka

- When a user logs on to the cluster, their log on to the head node of the cluster. Now, how do you run your calculation?
    i. Run ./your_application (wrong because (1) you're running computationally intensive tasks on the head node (2) You're not taking advantage of computing nodes' power
    ii. Login(ssh) into one of the compute nodes and run ./your_application (wrong because you do not know beforehand which computing node is open, you do not know if another user logs into the same node and runs intensive calculation, causing the node to potentially crash)
    iii. Request resources via scheduling software (right - scheduler is designed to schedule the resource requests for user jobs)

9. Chris Olson
    - How would you remove a commit in git from your history?
        i. Cherry pick the bad commit - indicates misunderstanding of what cherry pick does - it includes rather than excludes commits
        ii. Checkout the commit before, and cherry pick commits after - indicates misunderstanding of what checkout does
        iii. Create a new branch with a head before the bad commit and rebase interactively, skipping the bad one

10. Martino Sorbaro: Running *git checkout HEAD~1 file.txt* will make the file go back to a previous version and
    - A) *git status* return "nothing to commit"
    - B) *git status* display file.txt as "to be committed"
    - C) *git status* display file.txt as "modified"

This will make me understand whether the role of "checkout" is understood by the students. Checkout does not stage or commit the changes it makes to the current directory. This is important, because I can use checkout to change what I see, without changing the history.

11. André Palóczy

    Q: Which of the following code lines for calculating the sum of the cubes of two numbers 'a' and 'b' is _not_ correct?

    a) a**3 + b**3 ["wrong" (i.e., correct), shows that learner knows how to code exponentiation]

    b) (a + b)**3 ["correct" (i.e., wrong), indicates fundamental conceptual error in algebra, not programming]

    c) a*a*a + b*b*b ["wrong" (i.e., correct), shows that learner understands that exponentiation can be written as multiplication]

12. Anne Glaser

    How do you remove rows containing NA in your data frame?
    1. cats <- na.omit(cats)
    2. cats <- cats[-"NA",]
    3. cats <- cats.omit("NA")
        i. Identify their understanding of the use of functions with rows. Do they understand the difference between removing a specified row vs removing rows with specific data.

13.

14. Billy Broderick: You perform a convolution with the kernel [0 1 1 0] on the signal [1 2 3 4 5], with valid boundary handling. What is the output?
    - [5 7] -- correct
    - [0 1 3 5 7 9 5 0] -- doesn't understand the boundary handling (this is with full boundary handling)
    - [2 3 4] -- doesn't understand how summing happens with convolutions (these are the values that go into the correct output, but they should be summed in a pairwise fashion)

    - [12 20] -- confusion about order of operations, when summing and multiplication should happen (instead of multiplying between kernel and signal, then adding those outputs, here they added between kernel and signal, then multiplied)

15. Dima Lituiev: Categorical variables (factors) are useful representation of:
    - any textual data
    - variables with a small pre-defined number of categories
    - variables with a large number of categories that might increase in number
16. Craig Boman
    - How do I trim files in command line?
        i. SED - sed can edit files in place and make backups
        ii. CUT - cut does not edit files in place, only to standard output
17. Juliana Velez Lujan: Python supports the following data types but:
    •boolean:
    •integer:
    •long:
    •None: Is a distractor in itself, suggesting there is not a correct answer.

18. Julia Piaskowski
    - How do you update R?
      a) use the RStudio update function in “Help” tab
        -wrong answer - this updates rstudio, not R itself. Learner does not understand R and Rstudio are separate
      b) download the latest version of R and manually install it
        - Correct answer: indicates understanding of difference between R and RStudio
      d) run installr from RStudio
        - Wrong answer: shows understanding of difference between R and RStudio and comfort with command line, but lacks knowledge of detail on how implement this particular function
19. Nohemi Huanca-Nunez
    Q. Is it selecting 1st and 5th thru 10 variable?
    - a)newdata <- mydata[c(1,5:10)]
    - b)newdata <- mydata[c(1,5,10)]-

20. Saranya Canchi
    - How can one copy files in unix ?
        i. Use mv and cp
        ii. Use mv only
        iii. Use cp only
        iv. Neither mv or cp will work

Book with examples of formative assessment techniques:
Classroom Assessment Techniques by Angelo and Cross
https://www.amazon.com/Classroom-Assessment-Techniques-Handbook-Teachers/dp/1555425003

Formative Assessments Should be Frequent

How Many? (5 min)

Software and Data Carpentry use formative assessments often. How many have we done since the start of this workshop? Put your guess in the Google Doc along with one example and the purpose that assessment served.

- Dima Lituiev: 5
-
-
-

Key Points:

- Our goal when teaching novices is to help them construct useful mental models.
- This requires practice and feedback.
- Formative assessments provide practice for learners and feedback to learners and instructors.

III. How Learning Works: Expertise and Instruction

What is an Expert?

Name someone that you think is an expert (doesn't matter what they're an expert in). As an expert, what makes them special or different from other people?

- Katie Coburn: Jack Vevea. Very familiar with the "shape" of concepts -- able to quickly make connections
- Mara Sedlins: Yuichi Shoda (grad school advisor): able to identify the "meta" issues in the field of research (social psychology) and methodological approaches and articulate them clearly and in multiple ways
- Jacob Deppen - Seungki Kwak (colleague, residue analysis expert) - spent many days practicing techniques; learned from established experts; able to take on challenging problems; can explain his work to both novice and expert audiences
- Adriana Signorini - Linda Suskie- practical ideas - problem solver - idea connections -
- Elizabeth Salmon. Sara DS (colleague)brings experience and knowledge to correct, assist others; ability to critically evaluate and improve own work and colleagues;
- Matthias Bussonnier: Guido Van Rossum – created Python. They have long time experience in Python, but also historical involvent as to Why things are the way they are.
- Jeffrey Weekley Larry Smarr. Larry is able to make connections between highly disparate scientific domains. He engages in translational and transformational, as well as deeply impactful research in many areas.
- Sarvani Chadalapaka-Prof. Jason Losh (Teaching Embedded Systems) - they have deep thorough knowledge in the subject and are able to impart that knowledge to novices
- Chris Olson: Dave Sandwell - been processing satellite altimetry for years, taught the subject to many other successful scientists, and the products of his work have been used by many others
- Martino Sorbaro: Geoff Hinton - world expert in machine learning, one of the founders of the field, has a complete, bird's eye view of the field and its future directions and philosophical implications. -- An expert needs to learn from cutting-edge research or actual practice -- textbook knowledge is not sufficient anymore for further advancements of their knowledge.
- André Palóczy - Walter Munk (oceanographer) has done research in several sub-fields for ~80 years and had several original ideas, many of them very simple in principle, and designed ground-breaking experiments.
- Anne Glaser: Jaman Burton: Ability to explain complicated topics in a very simplified manner that anyone can relate to.
- Billy Broderick: Eero Simoncelli. In addition to a deep understanding of scientific and mathematical concepts, he's very good at conveying the intuition behind results / a methodology in a way that allows novices to follow along even if they don't understand the details.
- Dima Lituiev: Ivo Sbalzarini; understanding and ability to explain concepts in applied computer science.

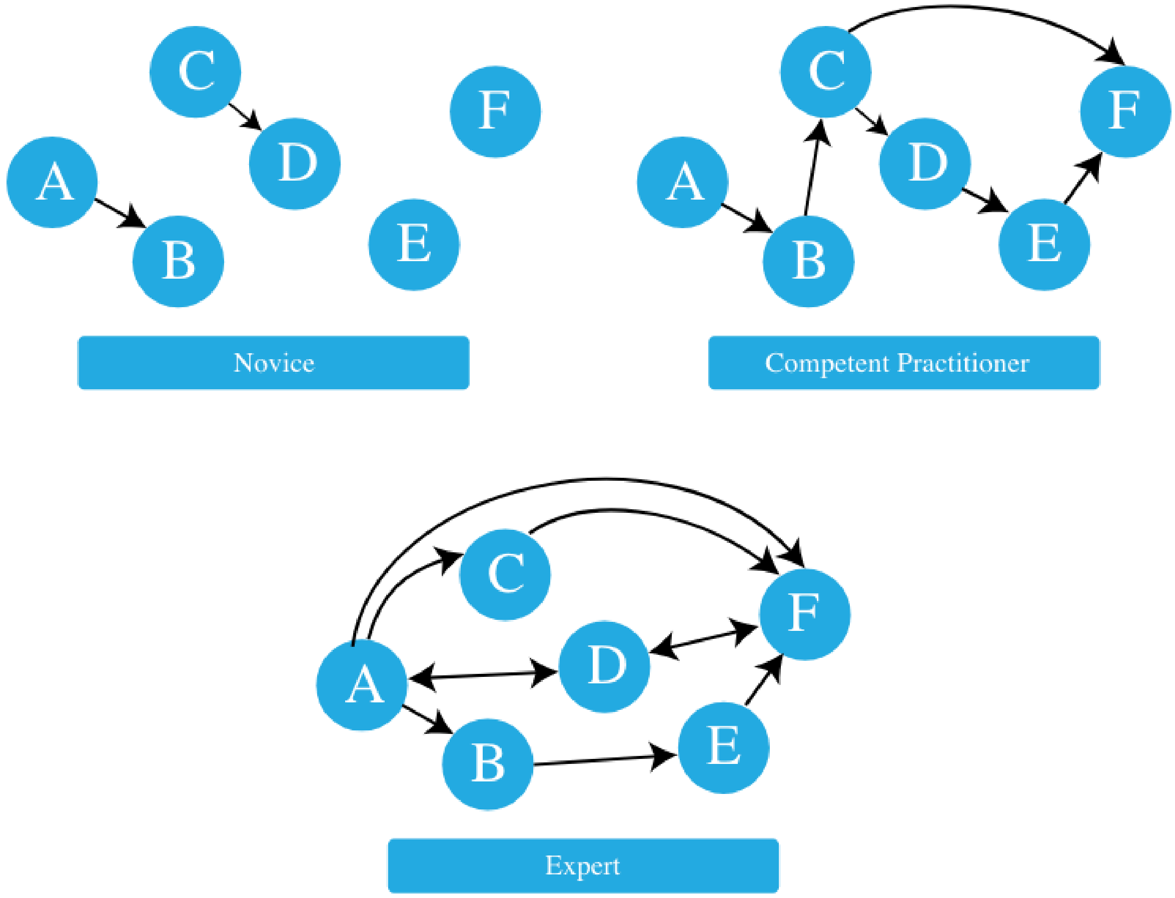
- Craig Boman - Linus Torvalds
- Juliana Velez Lujan: Carl June. Driven, passionate about his science and patients.
- Julia Piaskowski: Adam Liptak (SCOTUS, been on that beat for NYT for *years* - extensive knowledge and understanding)
- Nohemi Huanca-Nunez: John Terborgh, ability to explain complicated topic in simple terms
- Saranya Canchi - David Robinson - In depth understanding of the subject matter and the ability to convey abstract concepts in a easy to understand framework with clarity
- Eva Sanchez: Mark Hildebrand, diatom expert. Passionate, has studied diatoms for more than 20 years, loves his work, great mentor for students and postdocs
-

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

- Katie Coburn: statistics -- teaching about statistical concepts is comfortable and familiar, easier to come up with conceptual examples/analogies. Having to teach about behavioral genetics (once) was very difficult & I had to look up a lot of basic definitions to even begin.
- Mara Sedlins: playing the violin in an orchestra -- I don't have to focus on the mechanics of playing the instrument or reading music; I can pay attention to more complex tasks like being musically expressive or listening to the other sections of the orchestra
- Jacob Deppen - expert in archaeology of Mallorca, Spain; When I'm an expert, I am able to see how new research fits within existing research debates and conversations; when I'm not, I struggle to figure out what is important or why it is important
- Adriana Signorini- the steps of the assessment process or lesson plans - active learning skills
- Elizabeth Salmon: search; searching for information across disciplines, languages, time periods; assist others and educate them to become independent researchers;
- Matthias Bussonnier : Python. I know different ways of doing the same things, and which ones have which advantages/ drawbacks and how to choose depending on the context. I can also distinguish bug in My code from bugs in underlying libraries and language.
- Jeffrey Weekley - Virtual Reality. I tend to be much more skeptical in areas where I am expert.
- Sarvani Chadalapaka-building research computing communities: I tend to identify potential resources and try to bridge gaps...
- Chris Olson: Database design and data/metadata structuring of geophysical information in order to optimize querying and subsetting
- Eva Sanchez: green microalgae (Marinichlorella kaistiae KAS603). When I teach about microalgae, I'm better than teaching about something else.
- Martino Sorbaro: analysis of neural data, simulation of neural networks -- confidence in relating to others when talking about these concepts
- André Palóczy - Regional oceanography of the southwest South Atlantic. Done my BS and MS research on oceanographic processes there.
- Anne Glaser: Past products i've managed. I have an easier time navigating through unclear paths because it's easier to see the outcomes, where if I don't feel like an expert and something's not 100% clear it's hard to see what could go wrong.
- Billy Broderick: analysis of functional magnetic resonance imaging data. Not only do I have a sense how to perform a variety of analyses, but also the types of experiments in which they'll be useful and the kinds of inferences they support.
- Dima Lituiev: Applied machine learning. I am more *fluent* in concepts I am operating. Can relate different concepts and show how they are connected. Have broad overview.
- Craig Boman - Academic library database management - I have to be more careful about what I say in my library. As an amateur nobody listens to me.
- Juliana Velez Lujan. Biochemistry. The confidence levels are different between these 2.

- Julia Piaskowski: R, genomic relationship matrices, sewing & alterations. More confidence in evaluating things related to those topics
- Nohemi Huanca-Nunez, Peruvian traditional dance, guiding in the protected area of Manu National Park
- Saranya Canchi - Image analysis. The level of confidence is different
-

Connections and Mental Models



Limitations of Expertise

Fluid Representations (3 min)

Give at least one example of a fluid representation that you use in your own work. If you can, also give an example of a fluid representation that might occur in a Carpentry lesson.

- Katie Coburn: notation for the mean in a sample -- X bar, M, mean? If you've been using M the whole semester, the class will get super confused if you suddenly write X bar. Carpentries: getting the first few rows of a data frame; experts might know they can write "head()", novices might think they have to select every row manually ("data[c(1,2,3,4,5),]").
- Mara Sedlins: understanding that the same dataset can be expressed in an Excel file or a CSV file; in a Carpentry lesson: understanding that View(dataset) in R does the same thing as clicking on the name of the dataset in the Environment window of RStudio

- Jacob Deppen - I use the term “survey” to refer to a particular type of archaeological fieldwork where researchers walk around the landscape and document artifacts, not the type of survey most people think of; In a Carpentry lesson, I can quickly and easily read documentation, so parameters and docstrings make a lot of sense to me.
- Adriana Signorini - feedback is like the paddles to a boat. Qualitative data analysis and arrangement-
- Elizabeth Salmon: knowing when it’s time to move onto a different database or search engine; knowing when a search is completed (for the time…) ; obtaining experience that allows you to know/trust what works for you in nuanced situations
- Matthias Bussonnier: Jupyter: “notebook” can represent many concepts: “So I send you this notebook by mail, open it with jupyter notebook, so when you run the notebook, the code run in the notebook and will give you a result”.
- Jeffrey Weekley When 3D modelling or creating visual effects, it really doesn’t matter which tool I am using. I can switch between common tools rather easily, so if a student is more comfortable in one tool than the other, I can switch for them and still do the same activity.
- Sarvani Chadalapaka: I create cluster accounts for users and give them a sample example code and tell them “you’re all setup now. You can run your calculations on the cluster”. That’s fluid for me, but I understand it might not be for others
- Chris Olson: Relational versus non relational database design; procedural versus object oriented versus functional programming design
- Martino Sorbaro: have a principled process for understanding any error I may encounter in a Python script, even if I haven’t seen it before
- André Palóczy - In Python, I have all my frequently used functions in a toolbox like most people develop over the years, and am able to use it toghether with standard python libs. In learning new vocabulary in a new language, I try to tie words to examples in movies, songs or books that I’ve been exposed to. It helps me remember it and also to understand any subtleties in its meaning and use it more naturally.
- Anne Glaser: troubleshooting other people’s errors in R helps me see other ways to handle problems. I’m still new to R, but can teach the basic introductions. As people ask questions it makes me look at things in different ways and find new ways to fix the same issue.
- Billy Broderick: understanding how the terminal, IDEs, text editors, jupyter lab / jupyter notebook, shell scripts, python scripts, etc. all relate to each other (and the differences between them, when one might be more appropriate) when writing and running python code.
- Dima Lituiev: e.g. categorical variables are normally included in machine learning / statistical models by one-hot encoding, which motivates them being treated as a separate data type. Knowing the way variables are used helps to understand why they are a certain dataset.
- Craig Boman - I tend to use car analogies when explaining complex systems and dependencies
- Juliana Velez Lujan. Using several acronyms in a clinical setting to explain clinical protocols and trials that are ongoing.
- Julia Piaskowski: positional versus named arguments in a function
- Nohemi Huanca-Nunez - common species names (e.g 2 names for the same plant species), use of many names for the same object in general is problematic
- Saranya Canchi - The approach to debug irrespective of the language I work with. If I run into an error, what are the things I will try to get some understanding of the problem and how to break down the problem into manageable steps. Also trying alternative methods to achieve the same output.
- Eva Sanchez: what are microalgae? Why are they important? Where do they live? Does the world need microalgae? How are they being affected by global change. Finally, explain my research with microalgae. For a carpentry lesson start simple, with the basics, give examples related to the work of the students, make them comfortable for asking all questions (“There are no wrong questions”)

Diagnosis (5 min)

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 Google Doc.

- Katie Coburn: “Error: object ‘____’ not found.” Don’t run into this personally much any more, but it will come up if you don’t use the correct case in R (which is very case-sensitive). Watching case tends to trip learners up. I’ve had students say, “Why can’t it just recognize what I’m asking for?” I tell learners that it’s case-sensitive and that it’s completely normal/common to make this kind of typo (I still do it myself). The more times you type the same thing, the more likely you are to make typos. (An opportunity to emphasize the usefulness of RStudio/scripts.)
- Mara Sedlins: Error on a particular line of code -- I was missing something small like an apostrophe; explain to learners that every character/detail of code is important for it to function correctly and that it’s normal to make mistakes
- Jacob Deppen - “Error: expected 2 arguments but received 4”
    - In the course of programming, it’s important to know what each step of your code is doing. Part of that is understanding what is the nature of the output or result of an operation. An error like this one is telling us that what we gave to the function is not in the shape or form we expected. Maybe the data is stored in a different structure than we expected. For example, perhaps we were expecting a tuple of two numbers, but that particular function actually gave us a tuple of four numbers.
- Adriana Signorini- content analysis - identifying themes or trends in data - comparing and contrasting codes.
- Elizabeth Salmon: no results; need to rethink search terms, check boolean, strategies, may need to seek more background on topic, more experimental searches to identify correct terms, might be searching in wrong location
- Matthias Bussonnier:
    - Import Error: People are in wrong virtual environment. Need to explain that which environment you are in is on a per-terminal, per-session basis. Unlike usual apps that you install system wide in your own computer, you can “see” what is inside the environment, and it can shadow what’s already there.
    - Pip install - permission denied – Oh’ im just going to `sudo pip install`. No you are going to erase system critical things. Use `pip install --user`.
- Jeffrey Weekley - Data structure and transformations from one tool to the next can have a huge impact on the downstream quality of the calculations or representations. A common error is when someone inadvertently transforms data (e.g. compresses it algorithmically) that will have a deleterious effect downstream. You then must show them the differences in well-formed data (often inside a file where they’d never think to look) and the data they are producing. You explain that often, when you go from one format to another, you have differences that cannot be undone and you have to emphasize that each step along the way requires consideration.
- Sarvani Chadalapaka
    - X11 connection rejected:: Either they have x11 forwarding disabled or they do not have right DISPLAY environment variable
    - Can’t find a file: PATH is not setup right in their bashrc or add the directory to their PATH
- Chris Olson: Type handling errors - a lot of scripting languages don’t really impress upon new programmers that data object have different types, because most time interpreters can juggle and figure things out, but when they can’t it’s annoying and you usually have to go way back
- Martino Sorbaro: “Updates were rejected because the remote contains work that you do not have locally. This is usually caused by another repository pushing...” -- This is because we cannot overwrite other people’s work. Whoever pushes first has priority, and it’s now up to us to resolve conflicts. This can be done by pulling first.
- André Palóczy `ImportError` in python. Some module is not installed. It can be a module that the learner has cloned from a GitHub repo and forgot to run the `setup.py` script for [in case the module is not

available in `conda` or `pip`] (i.e., search and install dependencies, compile any code bundled in the module and copy them to the right directory, etc).

- Anne Glaser: I frequently see an error that comes up specific to an internal process related to concurrent processes trying to write to the same place.. Instead of just telling support that there's a concurrent database error and only copying the error and sending it to them, I take time to explain what's trying to write to the database and why it can't be deleted so it can be fixed from the source issue and they can easily spot the error next time.
- Billy Broderick: trying to run something on the HPC and getting a complaint about "package not found" or "command not found" or the like. Getting familiar with setting up environments on the cluster and loading in the necessary packages took a while to get used to and is not intuitive for many people (especially if they've never thought much about their computing environment before). I explain the error by trying to walk people through the idea that, when you first log in to the cluster, basically nothing is there, it's an almost clean slate, so you have to tell it which programs to prepare. Often the python packages necessary are included in anaconda, but some (neuroimaging-related ones) are not, and then having to explain virtual environments is more complicated.
- Dima Lituiev: cannot load module 'blah_blah'. This usually means a package needs to be installed. The steps to solve it is first to try installing the module using standard package repositories (CRAN, PIP) and if it fails, search for github or other repositories with that name.
- Craig Boman - Missing python package dependency error. Often the error message may point to a package you need to install, sometimes not. How can I research this error to find more meaningful information? Stack[overflow, exchange] and related forums.
- Juliana Velez Lujan. I don't think is an error per se, but the use of acronyms make it hard for students or coworkers not familiar with the topic to understand and follow. I just go back to the basics or initial place and explain what do they mean and what is the source to refer to.
- Julia Piaskowski: "subscript out of bounds" (I'm trying to access an object element that does not exist, which likely indicates an upstream error - something is a vector when I expected another format or their was an error in creating a particular object, or maybe I cut-and-pasted some code that needs more adjustment)
- Nohemi Huanca-Nunez: Error of calling the wrong subset of the data, or uploading the data in the correct format
- Saranya Canchi - Package install errors because some dependencies are either absent or they are not updated to the version required by the package. Usually occurs on shared systems and depending of the package, I either use conda environments or request update from system admin.
- Eva Sanchez. Font size. Some students may be disabled and not be able to read well. Bigger/bold fonts can be used (Ask the audience if they need that). Installing packages: ask the students to come early/ one day before class starts).

Blind Spots (5 min)

Is there anything you're 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?

- Katie Coburn: Teach! Still need to think about compensating for my expertise blind spot(s).
- Mara Sedlins
- Jacob Deppen
- Adriana Signorini - discuss misconceptions about learning. For e.g.,mass learning as opposed to space learning.
- Elizabeth Salmon- finding/providing the appropriate level of detail in explanations when teaching
- Matthias Bussonnier: Borrow Rules in Rust programing language. "A lifetime is missing, you need to add one". Where ? How ? why ? What's the Syntax already.
-
- Jeffrey Weekley ANY programming task...no, ANY one...

- Sarvani Chadalapaka Github navigation
- Chris Olson: Learning how to dockerize applications. I need to wrap my head around how to effetively split up services within an application and orchestrate them (as well as a bunch of new syntax)
- Martino Sorbaro: Finding appropriate sources in the literature
- André Palóczy - Studying Julia in Exercism (https://exercism.io). My mentor and the docs have been helpful to wrap my head around concepts like multiple dispatch, macros and automatic code generation.
- Anne Glaser - Learning R & Python. Some basic concepts I frequently have to google to double check. Dataframes is a great example from earlier. I had to look it up before class to remember the difference between a dataframe and a vector so I could better explain it.
- Billy Broderick
- Dima Lituiev: Navigating institutional politics, how to relate and incentivise collaborators.
- Craig Boman - Learning to train neural nets using clean data.
- Juliana Velez Lujan
- Julia Piaskowski - kalman filtering, whole new vocabulary for existing concepts in statistics
- Nohemi Huanca-Nunez : Identify patterns or relationship between variables, do all in one simpler function.
- Saranya Canchi
-

Think about the area of expertise you identified earlier. What could a potential blind spot be?

-

Dismissive Language

Changing Your Language

What other words or phrases can have the effect of demotivating learners? What alternatives can we use to express this meaning in a positive and motivational way? In the Google Doc, make a list of demotivating words/phrases and alternatives.

- Katie Coburn
    - “Simply,” “just,” “easy,” “quickly,” “There’s probably no questions on X so let’s move on,” “I’m going to skip this step because it’s self-explanatory,” “Like I said before.”
    - Try to omit adverbs like simply, quickly, easily.
- Mara Sedlins: “You probably remember how to do X from when you learned it …” → “Some of us may have encountered X before, but we could all benefit from reviewing … ”
- Jacob Deppen
    - “I’m sure you’ve all seen X before…”
    - “Maybe some people have seen X before, but if you haven’t, X is…”
- Adriana Signorini- use growth language -
- Elizabeth Salmon: just, it’s straightforward, all you need to do is, easy enough- alternative: let’s practice together
- Matthias Bussonnier
- Jeffrey Weekley - Keeping the principles of andragogy in mind: avoid “should” . Adult learners need to be asked questions in a way that they form their own answers. Focus on practical problem solving and provide them the tools to achieve their goals (not more, not less).
- Sarvani Chadalapaka
    - Demotivating: “You should have known that already”
    - Alternatives: “Here’s something for you to know as a background…
    -
- Chris Olson: “All you have to do is…” versus “This tool can be really helpful if you are trying to…”

- Martino Sorbaro “you haven’t listened” “you’re going to fail if…” -- “maybe I didn’t explain this clearly enough”, “you’ll get there if you try this...”
- André Palóczy “This shouldn’t be taking so long...” -> “Let’s have the ppl that finished (formative assessment exercise X) help out others who are struggling”.
- Anne Glaser: “Don’t worry about that...”. Instead if someone’s stuck on something you should explain it to them. If someone’s asking for an answer it’s better to explain it then to leave them struggling.
- Billy Broderick: “Is anybody still working on this?” -> maybe circulate around (or otherwise check without asking if people have finished) and offer help / instruct other learners to help those who are struggling
- Dima Lituiev: “it is not that straightforward”, “it takes time to learn”,
- Craig Boman - “do you understand…?”
- Juliana Velez Lujan. Is a piece of cake! Let’s review and identify what are we missing and fix it.
- Julia Piaskowski: “you should already know how to do this”
- Nohemi Huanca-Nunez - “so hard to do that you will not understand, so just copy what I am doing ” alternative could be to ask more attention and tell them that may take more time to do this
- Saranya Canchi - This was already covered, just do this, very simple → Alternative ways could be break the problem down and get answers from the learner to guide them towards the right idea, or say this concept is confusing , let’s revise once more to clarify
- Eva Sanchez: “This will take a very long time”/ “we need to put more effort on this and it might take more time”/ ”Take your time”.

You Are Not Your Learners

Key Points:

- Experts face challenges when teaching novices due to expert blind spot.
- Expert blind spot: knowing something so well that it seems easy when it’s not.
- With practice, we can learn to overcome our expert blind spot.

# Break

IV. How Learning Works: Working Memory and Cognitive Load

Types of Memory

Test Your Working Memory (5 min)

This website (http://opencoglab.org/memtest1/#interactive_test) implements a short test of working memory. You will be presented with a variety of different symbols, each presented for only a very short time, and asked to select those that you remember from a second set of symbols. There are 21 steps to the quiz, after which you can skip over the survey to see your results.

Take 5 minutes to complete the exercise. What was your score? Write your answer in the Google Doc.

- Katie Coburn: 52
- Mara Sedlins: 49 a helpful strategy was giving verbal labels to some of the characters that I could repeat to myself silently (e.g. “house”), but this was easier for some symbols than others
- Jacob Deppen - 39 🐡🐡🐡🐡🐡
- Adriana Signorini
- Elizabeth Salmon 41

- Matthias Bussonnier: 59 (Not all the symbols were rendering in my browser), also did non-browser based version in https://carpentries.github.io/instructor-training/05-memory/index.html (17, 3 missing (book, key, shoe), +2 in the wrong position (not sure if that counts).
- Jeffrey Weekley 48 (16 / 20 on word test)
- Sarvani Chadalapaka 40 (non-web based 13/20)
- Chris Olson - 42 not all the symbols showed up)
- Martino Sorbaro - 59 (some wrong chars)
- André Palóczy 54
- Anne Glaser - 46 - had to retake, would've give test results
- Billy Broderick: 58. Not all the symbols were rendering in my browser, so I had a lot of rectangles with four character codes in them (e.g., "26D6") (Martino: Same for me, I could do it anyway - the codes were easy to remember, easier than some symbols for me) (Billy: agreed)
- Dima Lituiev: 46 -- i guess black squares were supposed to be other chars ;)
- Craig Boman - 39 (did not see all the characters)
- Juliana Velez Lujan: 39
- Julia Piaskowski: 51
- Nohemi Huanca-Nunez 50 Not all the symbols were rendering in my browser
- Saranya Canchi - 53
- Naupaka Zimmerman: 54
- Eva Sanchez 35

Strategies For Memory Management

Improving Short-term Memory with Chunking (5 min)

Repeat the memory exercise you did earlier, but this time, try to form short stories or phrases from the words you see. Write the number of words you remembered in the Google Doc. How does this compare with your first attempt?

- Katie Coburn: 53 (+1). I don't think chunking works very well for this exercise. It is really difficult to immediately come up with a story connecting symbols when they are visible for such a short time
- Mara Sedlins: I got at 51 (+2 points) this time - it was hard to create stories for some of the image sequences
- Jacob Deppen: 62! +23!
- Adriana Signorini 40
- Elizabeth Salmon: 53, +12 sometimes images appeared to fast to chunk
- Matthias Bussonnier 59/59 (was already chunking, using images)
- Jeffrey Weekley
- Sarvani Chadalapaka
- Chris Olson: 42 not all the symbols showed up)
- Martino Sorbaro 63 (but I was using verbal clues earlier already)
- André Palóczy 55 (+1 point, same as Saranya below)
- Anne Glaser: 50
- Billy Broderick: 65 (+7 points). I found it hard to create stories for most of them, I felt like the improvement largely came because I was seeing the same symbols over again.
- Dima Lituiev: 47 -> 50
- Craig Boman
- Juliana Velez Lujan. 40 (+1 point only)
- Julia Piaskowski: 61 (+10, chunking worked until things were thrown at me too fast)
- Nohemi Huanca-Nunez 60
- Saranya Canchi - 57 - score improved but I wasn't able to make any stories as the images went by very quickly

- Naupaka Zimmerman (60, +6)
- Eva Sanchez (From 35 to 40) (+5)

Formative Assessment

Concept Maps as Instructional Planning Tools

Concept Mapping (10 min)
Create a hand-drawn concept map for a part of a Carpentry lesson you would teach in five minutes (ie. the amount of material you would teach before doing a formative assessment). You can use the same subject about which you created a multiple choice question, or a different subject. Trade with a partner, and critique each other's maps. Are there any concepts missing in your partner's map that you would include? Are there more than a handful of concepts in your map? If so, how would you re-divide those concepts to avoid overwhelming your learners' working memory? Take 10 minutes to draw the concept maps and share with your neighbor. **Write "*done*" below once you have finished.**

- Katie Coburn - done
- Mara Sedlins - done
- Jacob Deppen - done!
- Adriana Signorini - done
- Elizabeth Salmon- done
- Matthias Bussonnier
  - Trying to do a mindmap of GitHub repositories: Origin vs your fork, vs your local clone and how they relate to each other. Not happy with it.
- Jeffrey Weekley Explained the factoring required for deriving the possible number of colors in bit-space encoding of color in digital imaging.
- Sarvani Chadalapaka
  - Tried to do a workflow for a new cluster user walkthrough how they navigate around the cluster
- Chris Olson - done
- Martino Sorbaro - done
- André Palóczy - done (classes in Python)
- Anne Glaser - done
- Billy Broderick - done
- Dima Lituiev
- Craig Boman - done
- Juliana Velez Lujan
- Julia Piaskowski: done (running lm() in R)
- Nohemi Huanca-Nunez: done
- Saranya Canchi - done - Tried to do the UNIX file system -> found it not very intuitive to translate to this type of diagram

Feedback in the CH example from https://carpentries.github.io/instructor-training/05-memory/index.html

I would be careful with this example, as in particular ['a','b','c'] will not behave as "abc" as a collection.
In particular ("" in "abc") is True as "abc" contain the empty character, while ("" in ['a', 'b', 'c'] is False).
I've seen this example convey that "abc" and ['a', 'b', 'c'] were the same. Here it's a for loop, so it won't change, but is a dangerous stone to rely upon for collections.

Other uses of concept maps:

Why Guided-Practice is Important

Faded Examples

```
# total_length(["red", "green", "blue"]) => 12
def total_length(words):
        total = 0
        for word in words:
            total += len(word)
        return total

# word_lengths(["red", "green", "blue"]) => [3, 5, 4]
def word_lengths(words):
        lengths = ____
        for word in words:
            lengths ____
        return lengths

# concatenate_all(["red", "green", "blue"]) => "redgreenblue"
def concatenate_all(words):
        result = ____
        for ____ in ____:
            ____
        return result

# acronymize(["red", "green", "blue"]) => "RGB"
def acronymize(words):
        ____
```

Create a Faded Example from a Lesson (10 min)

The following exercise should be done in groups of 2-3.

1. Pick a block of code from an existing Software or Data Carpentry lesson, or from another lesson you have taught recently.
1. Replace 2-3 pieces of the code with a blank.
1. Write a question to test the student's ability to correctly fill in that blank.
1. Paste your faded example in the Google Doc.

Summary

Key Points

- 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.

- Teaching consists of loading short-term memory and reinforcing it long enough for items to be transferred to long-term memory.
- Use formative assessments to avoid overloading short-term memory.

V. Building Teaching Skill: Getting Feedback

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

Article on results from survey-based assessment of Carpentry workshops:
Short-format Workshops Build Skills and Confidence for Researchers to Work with Data
KL Jordan, M Corvellec, ED Wickes, NB Zimmerman, JM Duckles, TK Teal
2018 ASEE Annual Conference & Exposition https://peer.asee.org/30960.pdf

Minute Cards (aka Sticky notes)

One-Up, One-Down

Give Us Feedback (5 minutes)
https://docs.google.com/forms/d/1U7X9l7N9fHDEi1ct_V_c2ws_Q1B5_zrDgXA3t4TUWjA
Write one thing you learned this morning that you found useful on your blue sticky note, and one question you have about the material on the yellow. 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.

# Lunch

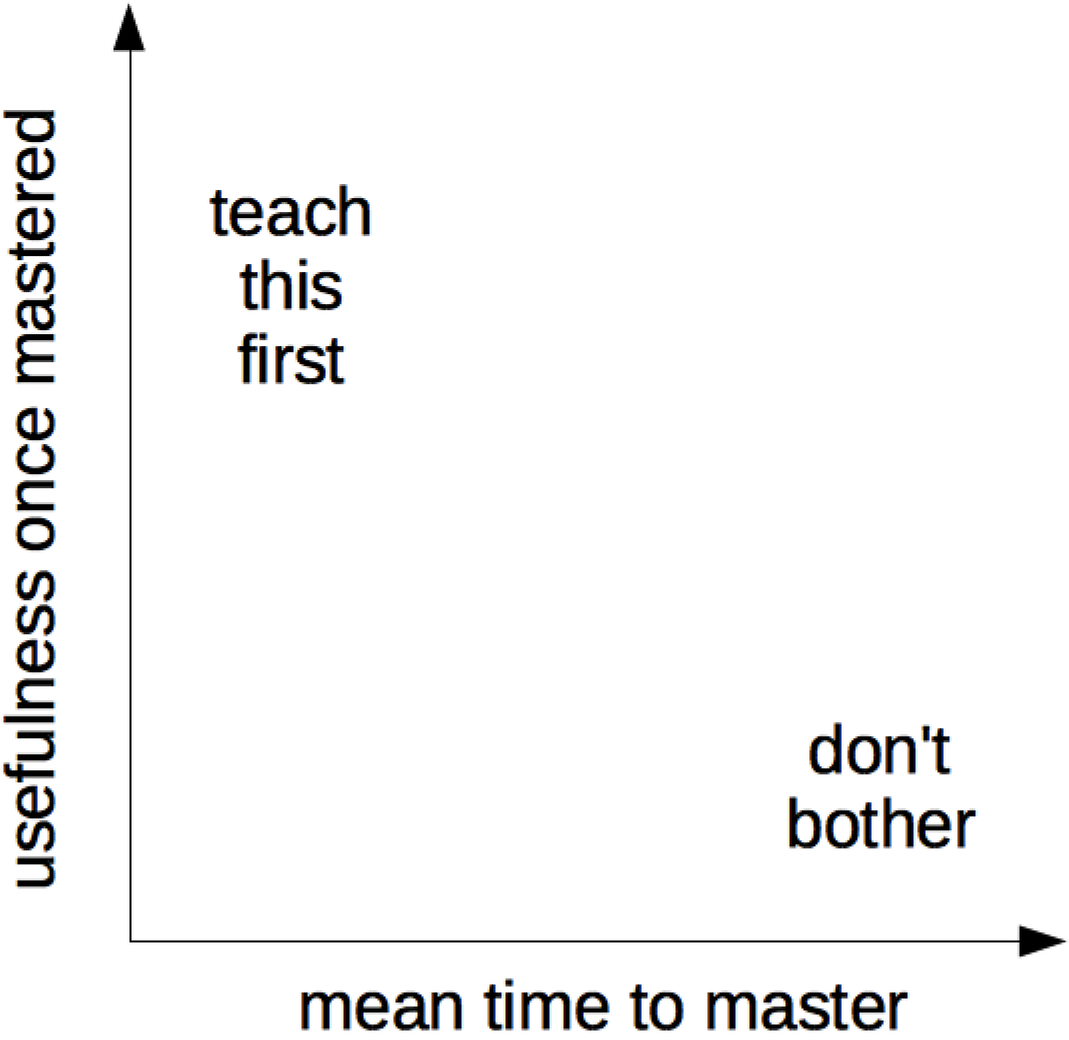
VI. Creating a Positive Learning Environment: Motivation and Demotivation

Badass book recommendation: https://www.amazon.com/Badass-Making-Awesome-Kathy-Sierra/dp/1491919019

Creating A Positive Learning Environment

- *Presenting the instructor as a learner*.
- *Establishing norms for interaction.*
- *Encouraging students to learn from each other*.
-
- *Acknowledging when students are confused.*

Teach Most Useful First



Authentic Tasks: Think, Pair, Share (10 min)
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, did some stats 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. **Pair** up with your neighbor and decide where this exercise fits on a graph of “short/long time to master” and “low/high usefulness”. In the class Google Doc, **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.

- Katie Coburn: wrote an R function to add to my R package (weightr). High on time to master (including debugging), but also very high on usefulness
- Mara Sedlins: used trinket.io to share some Python code along with instructions for how to use it; could be a useful tool to help learners practice writing documentation; relatively easy to use, potentially high usefulness

- Jacob Deppen - taking a small, simple dataset and turning it into a scatterplot or barchart is a really useful and quick to master skill (in the sense that mastering it means creating a plot with default styling that can help understand the nature of the dataset)
- Adriana Signorini
- Elizabeth Salmon - library related :) using RefWorks add-on in Google Docs- high on usefulness, low on time.
- Matthias Bussonnier,
  - (editfile, switch to terminal, run program, error, reapeat) to speedup that use http://eradman.com/entrproject/ to automatically run smth when you same any file in a directory.
- Jeffrey Weekley We do data transfers as a research IT service, but we prefer to teach people the "tricks" of how to do big data transfers successfully.
- Sarvani Chadalapaka At graduate research orientation we teach the basics of how to get onto our HPC resources. This is the most useful thing (to start).
- Chris Olson: awk + sed + |, how to access data through RESTful API, git is awesome even if you only learn 4 commands
- Martino Sorbaro: Git case: add and commit are easy and get you started quickly -- advanced commands (rebase, stash, bisect) are unnecessary at the beginning
- André Palóczy: Fast and useful: Intuitive iteration syntax in python: a = [column.mean() for column in table]. Slower but very helpful: pandas and xarray <3.
- Anne Glaser: Building documentation in markdown on how to use git from cli for beginners. Useful longterm, but difficult to learn up front for people who aren't familiar with git or the cli.
- Billy Broderick: writing documents with Latex. Takes a fair amount of time / effort (a very different mental model than writing things in word -- why does a document of text have to compile?), amount of payoff depends on complexity of document, amount of math, number of references, particularity of formatting. But in general, lower-right quadrant of that graph.
- Dima Lituiev: Basic git commands (clone, add, commit, push); Text processing: splitting on whitespaces, stripping punctuation.
- Craig Boman - Connecting Google sheets apps scripting to an external REST api call (useful but requires some mastery to tie it all together)
- Eva Sanchez: Create gene expression heatmaps using R. Teach this first to students that perform transcriptomics. It will help for a better understanding/mental model of the gene expression patterns
- Juliana Velez Lujan. Generate plots in R.
- Julia Piaskowski: how to analyze a split plot experiment in R, 4 different approaches and the differences between them. Provide example data and ask for their interpretation of the main effects. Medium effort to learn, very useful
- Nohemi Huanca-Nunez
- Saranya Canchi - Tidy way of doing R. Very useful and powerful to do many layered operations and the code is very easy to read

Strategies for Motivating Learners (5 min)

How Learning Works by Susan Ambrose, et al, contains this list of evidence-based methods to motivate learners. In groups of two or three, pick three of these points and describe in one sentence in the Google Doc how can we apply these strategies in our workshops.

Strategies to Establish Value

1. Connect the material to students' interests. Use example datasets that are relevant to the learners' field (in case of a very homogeneous class); Strategies 1, 2 and 3 **(UCSD group)**
2. Provide authentic, real-world tasks.And applicability of what is being taught.
3. Show relevance to students' current academic lives.
4. Demonstrate the relevance of higher-level skills to students' future professional lives.
5. Identify and reward what you value.
6. Show your own passion and enthusiasm for the discipline.

Strategies to Build Positive Expectations

7. Ensure alignment of objectives, assessments, and instructional strategies.
8. Identify an appropriate level of challenge.
9. Create assignments that provide an appropriate level of challenge.
10. Provide early success opportunities.
11. Articulate your expectations.
12. Provide rubrics.
13. Provide targeted feedback. (jdweekley) Use coaching techniques, rather than pedogogy you learned before.
14. Be fair.
15. Educate students about the ways we explain success and failure.
16. Describe effective study strategies.

Strategies for Self-Efficacy

17. Provide students with options and the ability to make choices.
18. Give students an opportunity to reflect.

- Katie Coburn **(from UC Merced Group)**:
    - #3 Could have students express their goal in broad terms prior to attending workshop and fuel your lesson with their examples.
    - #15 Demonstrate real-time debugging process; failure is part of the expectations in programming and success is a way of handling failure.

- Mara Sedlins - #1: Use example datasets from real research projects, ideally from the same discipline as attendees, #7: Spend time beforehand articulating your objectives and planning for assessment and instructional strategies, #10: Build in "early wins" as we discussed, designing activities that fall in the "teach this first" section of the graph above.

- Jacob Deppen - #6: I think most of us, by nature of wanting to be instructors, have a lot of passion and enthusiasm for these topics, so that is an easy place to share and encourage learners. Sharing your story and your own learning path. In my case, it is not a typical path of someone who learns to program, but programming has become invaluable to my work, so I would love to share that with people

- Adriana Signorini #16 effective study strategies. asks the class about any burning questions and their fears.

- Elizabeth Salmon #2 request examples from students based on their projects/research to use as tasks #18 reserve time for reflection/practice (and honor that time even when time is cramped)

- Matthias Bussonnier

- Jeffrey Weekley: Use coaching techniques, rather than traditional pedagogy.

- Sarvani Chadalapaka

- Chris Olson: #1, #2, #3 ask for a snippet of data from one of the students and performe a common task on it with the tools being presented. #4, #5, #6 show something that you are working on, expalian how you did it, and make it accessible to them. Along the lines of strategies to build positive expectations, I think giving an overview of why we are learning these tools and what we should be able to do with them by the end of the course is very helpful to do right off the bat. Often when I find myself getting lost in a lesson its rooted in not seeing the significance of why the information is being given to me.

- Martino Sorbaro: #2 and #3 ask students to provide simple datasets, and show how to plot and do simple analysis. #17 some exercises could be targeted to specific groups, so that students can choose their own preferred area of application (hard to implement at very basic levels, but great later on)

- André Palóczy # 1, 2, 3. Bring a small real dataset in the learner's field and explore it in the examples and assessments during the class.

- Anne Glaser: #8,9,10 Assign assignments that focus on basic skills for novices that may not know the basic concepts within R so they can build their skill sets with the fundamental building blocks instead of assuming they already have that knowledge or clearly understand it.

- Billy Broderick: #1, demonstrate usefulness of git / version control by referring to the classic folder full of "draft1.docx", "draft2.docx", "final_version.docx", final_version_v2.docx", etc

- Dima Lituiev: #1 Ask about whether they had experience managing coding projects, #2 show how large projects can grow and that git is necessary to manage such projects (e.g. pandas github page). #18: Ask them how they will use git next in their practice.

- Craig Boman: 11, 12 and 15, being clear about what success looks like and how you'll be evaluated in a rubric; for example clarity and commenting in code

- Juliana Velez Lujan: Provide authentic, real-world tasks.And applicability of what is being taught.

- Julia Piaskowski: Incorporate real and/or relevant data into examples that illustrate target principles. Establish learning objectives and use that to unify and align the parts of your workshop. Provide guidelines for what constitutes understanding of a topic - this provides a rubric and feedback.
  Nohemi Huanca-Nunez

- Saranya Canchi - I use anecdotes from my learning experience to highlight probable errors or mistakes while learning a new code / function / command

- Eva Sanchez:
  1. Connect the material to students' interests: Give examples related to their fields of interest.
  2. Provide authentic, real-world tasks: Analyze a worldwide problem that the entire class can relate to.
  3.Show relevance to students' current academic lives: Work with their grades, future plans

Brainstorming Motivational Strategies (5 min)

Think back to a computational (or other) course you took in the past, and identify one thing the instructor did that motivated you. *Pair* up with your neighbor and discuss what motivated you. *Share* the motivational story in the Google Doc.

Eva Sanchez: Slowing down and giving us time to think rather than memorize

Why Do You Teach? (5 min)

We all have a different motivation for teaching, and that is a really good thing! SWC 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.

- Katie Coburn: I’ve had great experiences with teachers and mentors throughout my lifetime, and I would love to be a great teacher and/or mentor to others in the future. I also find teaching to be one of the most rewarding things I’ve ever done and one of the best ways to make a difference in the world. Helping a student achieve that moment of connection, or that “click,” is a beautiful thing.
- Mara Sedlins: I want to give people the tools to engage in reproducible research more effectively and ultimately to change the culture of research to be more open, inclusive, and collaborative (e.g. by using tools like R Markdown to document and share analysis scripts).
- Jacob Deppen - The Carpentries tutorials on the shell and git really took my work to an exciting level I never expected. It helped me realize there is a lot of untapped potential in folks, so I want to help them find that in their work.
- Adriana Signorini Make faculty’s work more effective by sharing evidence based teaching activities. I learn at the same time!
- Elizabeth Salmon: Love to share my enthusiasm and curiosity and hope to instill it in others and help them grow into into effective and independent researchers
- Matthias Bussonnier
    - Love to be able to make other people more efficient and more independant. Typically I love to make myself useless, so that I can move onto helping with something else.
- Jeffrey Weekley I teach because I enjoy learning. I’m rarely an expert in the topics I teach - I am more of a generalist, but in teaching, I always learn (or relearn) something useful.
- Sarvani Chadalapaka
    - I am inspired by all the awesome teachers I have had. Besides, I want to understand the broader impact of what I do and teaching can help me figure out what I should know to encourage more people to use the cool technology I use.
- Chris Olson: I often hear very smart people express that they are very intimidated by coding, when I know that it can really help their work, they just may not know where or how to start. I love explaining things as it really solidifies/checks my own understanding of topics that I am presenting. I also aspire to be like the mentors that taught me.
- Martino Sorbaro: Better understanding of the contents myself; feeling rewarded, powerful, at the centre of attention; interacting with other people (as opposed to research which is a lonely activity); feeling it’s an activity that has an immediate impact on other people and the external world.
- André Palóczy - I’ve only had awesome mentors since sophomore year in college, and have come to really enjoy mentoring and take it seriously. This motivates me to want to get better at teaching and mentoring.
- Anne Glaser: Most people just teach others how to complete something and skip why it happens that way. I find it helpful to take the time to teach people the background on what’s happening in a process to cause a problem, not just focusing on the steps to fix it and moving on.
- Billy Broderick: passion for the material and seeing that light behind someone else’s eyes when they understand something / also think something is really cool. For programming-related things specifically,

they're so important for doing reproducible and cumulative science, and scientists are very rarely taught them in any way.
- Dima Lituiev: share what makes my life easier and more enjoyable, make people feel good about their abilities.
- Craig Boman - to ignite a passion for learning, rather than a specific topic
- Juliana Velez Lujan: I like to provide access to knowledge to different communities. I like to be an example of progress and opportunities. I like to empower others.
- Julia Piaskowski: to create more collaborators, to improve the quality of data processing and data analysis, to give back to my community
- Nohemi Huanca-Nunez
- Saranya Canchi - I like to teach to become a better learner myself.
- John Simpson - I learn a lot teaching. Often more than I do in other situations.
- Eva Sanchez: I like to teach because it is a great way to help, motivate and develop critical thinking for students. This will help with their personal lives, careers and problem solving. In the end it can make students happier and able to help the world. Teaching is also a great way to learn, I consider it as team work.

How Not to Demotivate Your Learners

Brainstorming Demotivational Experiences (5 min)

Think back to a time when you were demotivated as a student (or when you demotivated a student). *Pair* up with your neighbor and discuss what could have been done differently in the situation to make it not demotivating. *Share* your story in the Google Doc.

- Katie Coburn
- Mara Sedlins
- Jacob Deppen
- Adriana Signorini
- Elizabeth Salmon
- Matthias Bussonnier
- Jeffrey Weekley
- Sarvani Chadalapaka
- Chris Olson
- Martino Sorbaro
- André Palóczy
- Anne Glaser
- Billy Broderick
- Dima Lituiev
- Craig Boman
- Juliana Velez Lujan
- Julia Piaskowski:
- Nohemi Huanca-Nunez
- Saranya Canchi
- Eva Sanchez: creating a stressful environment can be demotivational

Psychological Demotivators

Stereotype Threat

Impostor Syndrome

Overcoming Imposter Syndrome (5 min)
Think of a time when learning something was difficult for you, or you made a mistake that seemed silly or embarrassing. Is that task still hard for you? In the Google Doc, describe how you might you use this as a motivational story to help your learners overcome their own imposter syndrome.

- Eva Sanchez: Stress and lack of sleep
- Adriana Signorini: share initial struggles with the discipline/ as an immigrant
-
-
-

Accessibility Issues
Ask in a survey what is students are apprehensive about and want the instructor account for.

One Thing (5 min)
The UK Home Office has put together a set of posters (https://github.com/UKHomeOffice/posters/blob/master/accessibility/dos-donts/posters_en-UK/accessibility-posters-set.pdf) of dos and don’ts for making visual and web-based materials more accessible for different populations. Take a look at one of these posters and put one thing you’ve learned in the Google Doc.

- Katie Coburn
- Mara Sedlins
- Jacob Deppen
- Adriana Signorini
- Elizabeth Salmon
- Matthias Bussonnier
    - I just want to add a note that disability can be hard to spot, or temporary:
    - - Foreign Speaker, Broken Arm, In the back of the room with sun shining in your eyes because there is no curtain...
- Jeffrey Weekley
- Sarvani Chadalapaka
- Chris Olson
- Martino Sorbaro
- André Palóczy
- Anne Glaser
- Billy Broderick
- Dima Lituiev
- Craig Boman
- Juliana Velez Lujan
- Julia Piaskowski
- Nohemi Huanca-Nunez
- Saranya Canchi

What Happens When Accessibility is an Issue? (5 min)
Think of a time when you’ve been affected by, or noticed someone else being affected by issues with 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

-
-
-
-

-

Inclusivity

Key Points

- A positive learning environment helps people concentrate on learning.
- People learn best when they see the utility in what they're learning, so teach what's most immediately useful first.
- Imposter syndrome is a powerful force, but can be overcome.
- Accessibility benefits everyone.

VII. Creating a Positive Learning Environment: Mindset

https://carpentries.github.io/instructor-training/09-mindset/index.html

Does Mindset matter? (5 min)
Think: What kind of mindset do you have about different areas? Is there anything you believe you are "not naturally talented" at? Mindset often varies in different areas – someone might have a fixed mindset with respect to artistic ability, but a growth mindset with respect to computing skill. Then, think about your learners. How might a learner's mindset about computational skill influence their learning in a workshop setting?
Pair: Discuss your thoughts about the influence of mindset in a workshop. Try to come up with a few different ways or situations in which mindset might be relevant.
Share: A few thoughts in the Google Doc (or go around the room and discuss)

Praise Influences Mindset

Choosing our Praises (5 min)
Since we're so used to being praised for our performance, it can be challenging to change the way we praise our learners.
Which of these are examples of performance-based, effort-based, or improvement-based praise?

- I like the way you tried a couple of different strategies to solve that problem.
  - Effort-based
  - effort-based
  - Effort-based
  - Improvement-based
  - Effort based
  - Performance based
- You're getting really good at that. Keep up the hard work!
  - Effort-based
  - Improvement-based
  - Improvement-based
  - Improvement Based
- You're really talented.
  - Performance-based
  - Performance-based
- That was a hard problem. You didn't get the right answer, but look at how much you learned trying to solve it!

    - Improvement-based
    - Effort-based
    - Effort Based
    - Improvement based
    - Effort-based
- I see how much time and effort you put into this!
    - Improvement-based
    - Performance based
    - Effort-based
    - Effort-based
- That's a solution I would not have thought of myself. It's fantastic. Wasn't it worth the effort?
    - perfor

Errors are Essential to Learning

Helping Learners Learn From Mistakes (5 min)
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 Google Doc, describe the error your learner has made and how you would respond.

- Katie Coburn: Error -- the learner created a data frame called *cats* and tried to reference one called *Cats* (in R). "You did a really good job -- I think the only reason your code didn't work is that R is very case-sensitive. Do you remember what you called the data frame when you made it? [Also, assuming they're using RStudio] One other nice tool for checking this is the Environment pane up here."
- Mara Sedlins: Say, oh, I used to do this all the time! (if true) Then ask them what directory they want to look at and what command lets them look at the files, and try to direct them toward identifying the error on their own.
- Jacob Deppen - Let's try to find out why it isn't working first, then we'll try to fix it. Do you have a guess at what the issue is? Did you get an error message that might give you some clues? Given that we got a SyntaxError, what is the syntax we are trying to run?
- Adriana Signorini. Peer review session: Let's check each other's data frame and provide feedback..
- Elizabeth Salmon: Let's go back and review the initial instructions. Can you see the difference between this and your command?
- Matthias Bussonnier
    - Without looking at what you wrote, try to describe what you are trying to achieve and how.
    - Without thinking about what you are trying to achieve, describe what you have written.
    - Now try to find inconsistency between both.
- Jeffrey Weekley - Let's try that to see if it will work...OK, that means there's an error somewhere in your script/command/code. Let's take a look at this carefully to see if we can figure it out. It's not jumping out to you, so let's look back at the example and compare it with what we did before.
- Sarvani Chadalapaka
    - Ls command to list the files - you did that right. Now, what error are you seeing?
    - What do you think that error means?
    - How do you think we should fix this?
- Chris Olson: They probably got a file/command not found error - Why is bash looking for a file/command? What is the name of the file/command it is looking for?

- Martino Sorbaro: Let's have a look at the error message. Can we learn something from it? If not, let us compare again with what the lesson materials show. What's the difference? Let's try out the first hypothesis of what could fix it, and then repeat.
- André Palóczy: Walk them through the problem with two examples: "Let's do a test: Try typing in just "ls subdirectory-in-cwd/". Now try "lssubdirectory-in-cwd/". [error message] Why did an error pop up in the second one but not the second?"
- Anne Glaser: Try to start the problem from the beginning so that when they reach that point you can stop them and start asking some questions about why they decided to do that. That would help me understand their reasoning and find a way to explain to why it didn't work and help them figure out what would.
- Billy Broderick: (after they've seen it doesn't work) "can you walk me through the command here and your thought process?" try to find the fundamental error, correct that, and then ask them how they'd fix the command reflecting that new knowledge
- Dima Lituiev: ask them to guess what might went wrong. Point to an example that can help them to infer correct way to do it. If that does not work, explain what is correct and what is broken.
- Craig Boman: let's look at the manual for ls to see if there is some more information on formatting.
- Juliana Velez Lujan: I would ask, any idea what was the mistake you made? Have you reviewed carefully the code?
- Julia Piaskowski: ask learner to examine their output (does it look right?) or ask what the learner thinks the error message means, encourage them to check the help files
- Nohemi Huanca-Nunez
- Saranya Canchi - lets try to understand this error. Can you walk me through the command and what steps you are planning to do ?
- Eva Sanchez: Would you like to recheck your code? How about if you try again by adding a space here?
-

Ask to explain what they have done step-by-step (rubber duck method)

Perseverance Predicts Success

How Are You Gritty? (5 min)
A previous exercise asked you to think of a time when learning something was difficult for you, or you made a mistake that seemed silly or embarrassing.
How did you motivate yourself to continue learning? How did it feel to persist in the face of challenge? How do you feel now about your capabilities in this area?
In the Google Doc, describe how you could use this story to illustrate the importance of grit for your learners.

-
-
-
-
-

Habits of Lifelong Learners

Key Points

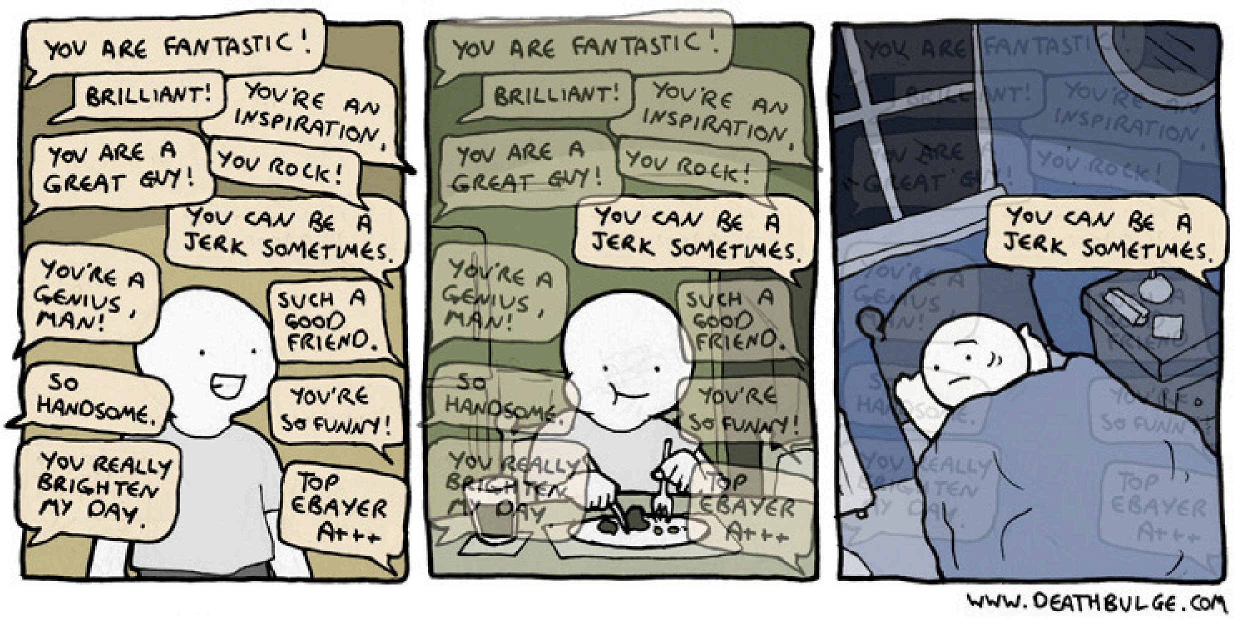
- Growth mindset and grit promote learning by making effort a positive thing.
- Presenting errors as essential to the learning process helps learners learn from their mistakes.
- Successful lifelong learners aren't embarrassed to ask for help.

Break

VIII. Building Teaching Skill: The Importance of Practice

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

Lesson Study: Applying a Growth Mindset to Teaching



Giving Feedback (10 min)
We'll start by observing some examples of Carpentry workshop teaching and providing some feedback.
Watch this video (https://www.youtube.com/watch?v=-ApVt04rB4U) as a group and then give feedback on it. Put your feedback in the Google Doc. Organize your feedback along two axes: positive vs. negative and content (what was said) vs. presentation (how it was said).

- Content +
    -
- Content -
    -
- Presentation +
    -
- Presentation -
    -

- Katie Coburn
    - Content +
        - Functions -- informative information, useful
    - Content -
        - Somehow managed to be both boring and confusing
    - Presentation +

- Live coding, reading each command aloud
- Pointed/gestured at relevant lines to emphasize
- Presentation -
    - Phrases like “this is really simple stuff”
    - Lot of “um,” “uh”
    - “Which of course is polymorphic” -- not intuitive to everyone
    - “For those of you who haven’t seen functions before”
    - Started back up basically yelling at them

- Mara Sedlins
  - Content +
    - picked up where they’d left off before, reviewing the previous example
  - Content -
    - seemed to be building on the material, in line with the “faded examples” type of approach (although it was hard to see what was on the screen)
  - Presentation +
    - made a mistake during live coding, which is a good teaching opportunity
  - Presentation -
    - didn’t capitalize on the opportunity to explain the mistake he made; was impatient with the classroom; got distracted by his cell phone; used phrases like “this is really simple,” “which of course …”; spoke quickly without offering opportunities to ask questions or check for understanding
-
- Jacob Deppen
  - Content +
    - Had a lot of expertise, showed some interesting uses of functions
  - Content -
    - Use of jargon and technical terms was at too high a level for the purposes of an intro workshop
    - Was little to no time for learners to ask questions or ask for help
  - Presentation +
    - Seemed excited about what he was teaching
  - Presentation -
    - Got distracted by his phone at one point
    - Was a bit harsh to a person in the audience

- Adriana Signorini
  - Content +

  - Content - no outcome setting

  - Presentation + He showed up! Teaching by doing and making mistakes and trying again.
  - Presentation - small font- not visible - like - not prepared - poor lighting

- Elizabeth Salmon
  - Content +
    - Used an authentic example
    - Really know your stuff!
  - Content -
    - Implied that this material was simple through dismissive language
  - Presentation +

        - Helpful when you moved over to point out things out on the screen
    - Presentation -
        - Some people need more of a warning when starting back up and a minute or two to get settled
        - Suggest slower pacing
- Matthias Bussonnier
    - Content +
        - Greg is a really friendly person.
        - Use accurate vocabulary.
        - Usefull _advanced_ topics.
    - Content -
        - Usefull _advanced_ topics (not for beginner)
        - Usuing dismissive words “Just saying what is do not why.”
    - Presentation +
        - All steps on screen.
        - Describe every step he does (for the first half), then swithc to “oh i’m just going to fix it”.
    - Presentation -
        - way too fast/ text too small.
        - rminal too small
        - stuttering/unclear
        - Too many concepts.
- Jeffrey Weekley

    Content +
    - Content: If the intent was to translate this training to video, then the lighting, the display and the speaker stagecraft could be improved.
    - Content: The speaker didn’t seem to give realistic pauses for the students to actually interrupt him.
    - Presentation. How do you feel that presentation went for you? Were there things you felt could have gone better? Can you identify areas where you might do things differently next time?
    - Presentation: There was enthusiasm, but maybe not the best presentation
- Sarvani Chadalapaka
    - Content +
        - Subject matter - clearly he is an expert in the subject

    How you feel that presentation went for you:
    - What just happened in that video ?!!!

    Things that could have gone better:
    - Font and pace - but honestly, I am glad it is a video because I could pause it and/re-watch it. If it is live, I would have absolutely been fumbling.

    Areas where you might do things differently:
    - Use of technical jargon like functions, invoke, pass it in, parameters
    - Not clear - use of terms “like that” (without giving a context and really explaining why he was doing what he was doing) and “trust me”
    - Not explaining thought process
- Chris Olson
    - Content +
        - Useful skills, addressing functions and data types
    - Content -
        - Condescending towards excel users
        - Glossing over mistakes
        - Assumed knowledge of certain things

  - Presentation +
    - Active, using hand gestures and pointing at what he is referring to
    - Seemed to make sure everyone was seated and paying attention
  - Presentation -
    - Checked phone generally scattered and interrupted
    - Using non laymen lexicon
- Martino Sorbaro
  - Content +
    - Provides examples
    - Basically nothing
  - Content -
    - Mixes basic and advanced stuff, jumps randomly from an dea to another
  - Presentation +
    - At least you won’t fall asleep
    - Live coding
  - Presentation -
    - Very confusing, speaks fast, says “very simple stuff”, throws random buzzwords (“polymorphic”), asks students to “trust him” instead of showing evidence

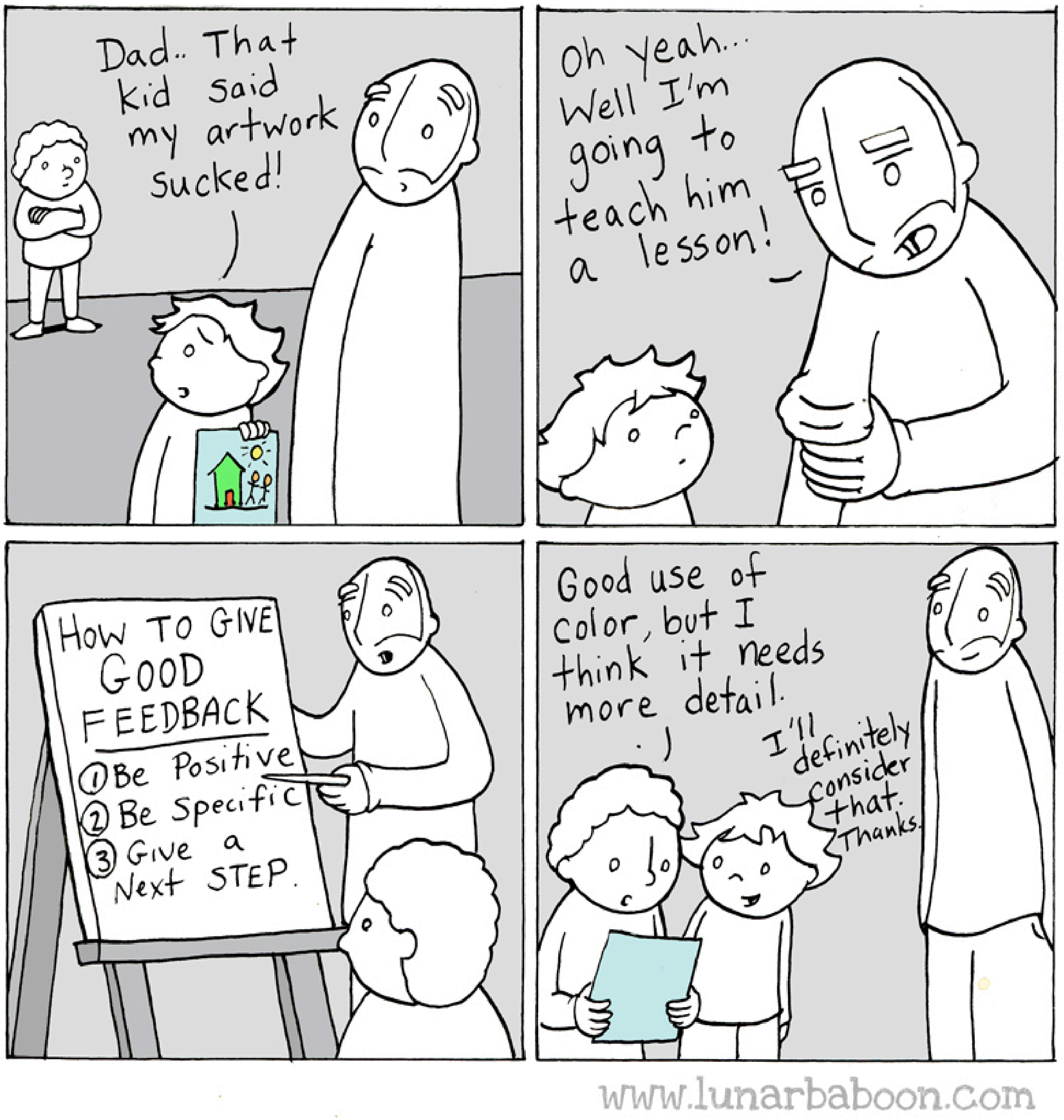
André Palóczy

- Content +
  - Explaining functions with live coding examples. Both useful and appropriate for novices.
- Content -
  - Polymorphic? Not appropriate or useful for intro level.
- Presentation +
  - Doing live coding is good. But it’s going too fast.
- Presentation -
  - Demotivating passages. Too fast. Font too tiny to read. Looking at his phone.
  - “Even an Excel user gets this”: Both demotivational to the class ‘e’ and against the Code of Conduct (stereotyping Excel users as less capable).
- Anne Glaser
  - Content +
    - Started with basics and built forward
  - Content -
    - He seemed rush through it, kept saying don’t worry about it
    - Kept repeating that it was simple.
  - Presentation +
    - Live coding
  - Presentation -
    - Difficult to see what he was typing, needed to zoom in on what he was working on
- Billy Broderick
  - Content +
    - Makes a mistake and then fixes it in front of class (but doesn’t explain thought process)
    - Examples seem like they build complexity (introduces function of variables and then function of functions)
  - Content -
    - Lots of jargon, referencing advanced topics while supposedly teaching a basic one
    - “Don’t worry about this, it’s what you’d expect”, not explaining concepts as necessary, overwhelming people
  - Presentation +
    - Live-coding, reads what he’s doing (even if he doesn’t explain it)

    - Presentation -
        - “This is simple, even Excel users can do it”, not only using dismissive language, but also insulting people who may be in the audience
        - Looks at phone in the middle of lecture
- Dima Lituiev
    - Content +
        - Gradual step-by-step explanation
    - Content -
        - some domain terms are not exaplined (e.g. ‘polymorphic’)
    - Presentation +
        -
    - Presentation -
        - ‘Just’, ‘even an excel user can understand this’
- Craig Boman
    - Content +
        - Seems to know what he’s talking about
    - Content -
        - Was not organized
        - Too much jargon
        - Assumed higher level of knowledge
        - Condescending
        - Didn’t explain goal of the explanation
    - Presentation +
        - Overcomes errors
        - Asked for questions at the end
    - Presentation -
        - Difficult to hear (spoke to the podium)
        - Didn’t speak clearly
        - Was rude at the beginning
- Juliana Velez Lujan:
  Presentation: Teacher seems exacerbated with a student and asked him/her to sit down.
  Content: He makes a comment about how simple an instruction is that somebody using excel could just know/do it.
  Content: “This is what you would expect, trust me”, doesn’t seem to me to teach much.
  Content: He is speaking and working too quick, he is fixing code without explaining further.
-
- Julia Piaskowski
    - Content+ could not evaluate
    - Content- unclear what he was talking about did not explain concepts or break content down into manageable parts
    - Presentation+ live coding
    - Presentation- rude to people, dismissive language, checked his phone, very small display font
- Nohemi Huanca-Nunez
- Saranya Canchi
    - Content +
        - Have none that would be considered a positive :(
    - Content -
        - Not defining functions and def, trying to create higher order functions without giving context, using arbitrary names like foo, stating “this is simple stuff”
    - Presentation +
        - Lots of cool laptop stickers

    - Presentation -
        - Very fast speech, lot of distracted movements, thoughts are not well organized (on the fly)
- Eva Sanchez
  Content: To complicated, lack of organization
  Presentation: Stressful, a little aggressive

Feedback on Yourself (25 min)


Dad.. That kid said my artwork sucked!
Oh yeah... Well I'm going to teach him a lesson!
HOW TO GIVE GOOD FEEDBACK
① Be Positive
② Be Specific
③ Give a Next STEP.
Good use of color, but I think it needs more detail.
I'll definitely consider that. Thanks.
www.lunarbaboon.com

1. Split into groups of three.
2. Individually, spend 5 minutes preparing to teach a 90-second segment of the Carpentry lesson episode you chose before the start of the training course.
3. Get together with your group and have each person teach their segment to the group, while one person records this (video and audio) using a cell phone or some other handheld device. Use a whiteboard or other visual aids if available, but do not use live coding. We'll be practicing with live coding later. Keep a strict time limit of 90 seconds per person.
4. After the first person finishes, rotate roles (she becomes the videographer, her audience becomes the instructor, the person who was recording becomes the audience) and then rotate roles again.
5. After everyone in the group of three has finished teaching, watch the videos as a group. Everyone gives feedback on all three videos, i.e., people give feedback on themselves as well as on others.
6. After everyone has given feedback on all of the videos, return to the main group and put everyone's feedback about you into the Google Doc.
    -

- Katie Coburn: Make sure to reiterate concepts throughout the lesson. (One tip: mention that sometimes language-specific vocabulary is necessary for researching solutions to errors.) Draw analogies when introducing new vocabulary terms.
- Mara Sedlins (feedback from Julia Piaskowski): what is a data frame? Plus: Good intro into why this matters, clear explanation that build on concept of vectors, good visual example, involved audience (me). Minus ran out of time - might need more time to cover this topic, or more narrow function -- Mara's summary of the feedback: I did a good job of introducing the context/importance of the topic but ran out of time to get to some important conceptual information, I used a visual aid to help illustrate the information and asked for feedback to help clarify understanding of the topic
- Jacob Deppen - In introducing pandas as a library, it would be good to include examples of what pandas can do for the learner *right now*. My examples were somewhat "future-focused" (i.e., here is what pandas can do at a more advanced stage), which might not have the same motivating power for novice learners.
- Adriana Signorini
- Elizabeth Salmon
- Matthias Bussonnier:
    - Expected Feedback: I speak too fast.
    - Given Feedback: I speak too fast. We can see I am excited about the subject.
- Jeffrey Weekley - It was conversational style. Took time out to answer questions.
- Sarvani Chadalapaka
- Chris Olson: I spoke about bash pipes. Be clear in defining the scope of when pipes are used - although they may exist in various forms in other languages, this is a bash thing. Also be clear that intermediate data isn't saved in the directory when doing this.
- Martino Sorbaro: interacted with the audience by using "we" "let us", making them do tasks; chosen an introductory topic; fast-paced; use examples of real use case, and/or analogies for harder concepts
- André Palóczy: Ran too long, need to cut down on material; need to speak facing the audience more. Nice presentation in comparing 'for' and 'while'.
- Anne Glaser: Explained basic concepts. Used excel as an example of a data frame. Kept using the word 'just', it's a hard habit to break.
- Billy Broderick (version control with git): did a good job going slowly, avoiding demotivating language and jargon. For next time, should spell out nano, explain how to quit, explain what cat is; more broadly, as a learner, it can be difficult to determine what's an arbitrary choice and what's necessary, so try to make that explicit.
- Dima Lituiev:
    - Showing contents of the file is helpful before reading the file in pandas

    - The head command needs to be explained
    - Variable assignment and dereferencing must be explained
- Craig Boman - useful to ask why we should use pipenv at the start; slow down, especially in cli; avoid quickly; not everyone can cd or mkdir quickly
- Juliana Velez Lujan:
- Julia Piaskowski (feedback from Mara Sedlins): I discussed what is a data frame in R. My feedback is that I covered a lot of important material in a short amount of time, related information back to common experience (comparing to Excel). It was suggested that I use specific examples of different data types in R, perhaps using examples from people's research to make this more palatable and intuitive.
- Nohemi Huanca-Nunez
- Saranya Canchi - Feedback from the UCSD group Good presentation and clear concepts. Could have explained the concept of pipe better and why it is useful compared to base R version of subsetting.
- Eva Sanchez. GitHub: Need to explain the importance more clearly, why should a student be interested in GitHub. How does it facilitate the life of a student? Why is it efficient for storing code?

Feedback Is Hard

Feedback on Feedback (15 min)

Watch either this video(https://vimeo.com/139316669) (8:40) or this one (https://vimeo.com/139181120) (11:42). For each, give feedback following the 2x2 model. Put your feedback in the Google Doc.

-
-
-
-

Using Feedback (5 min)

Look back at the feedback you received on your teaching in an earlier exercise. How do you feel about this feedback? Is it fair and reasonable? Do you agree with it?
Identify at least one specific change you will make to your teaching based on this feedback. Describe your change in the Google Doc.

- Katie Coburn: Draw more analogies when introducing new vocab terms. I definitely think that's reasonable and will try to incorporate analogies into lessons more often.
- Mara Sedlins: I would plan ahead more carefully to allow time to cover what I thought was important; actually, watching Julia teach (thinking about the material from the perspective of a learner) gave me ideas about my own teaching as well, e.g. bringing in different examples or doing a class activity that generates example data to work with
- Jacob Deppen - my reviewers had a very nice, clear suggestion that I could adopt: instead of discussing how pandas can be integrated with other more advanced libraries, tell students how pandas by itself can be really useful to them right away. Now I need to think of the perfect example for that.
- Adriana Signorini
- Elizabeth Salmon- feedback from previous sessions, speak more clearly, give examples
- Matthias Bussonnier
    - Breath, take a pause and drink regularly between my sentences to slow down the pace.
- Jeffrey Weekley - For some subjects, a conversational style is unnecessarily slow. For basic concepts, I would go faster to be able to cover the material in the time allotted.
- Sarvani Chadalapaka: Use a whiteboard, to try to use more program-specific terminology(syntax)

- Chris Olson: I agree with the feedback and it really highlights to me that there are many different perspectives and ways of looking at something than just my own. I need to practice more as well as check in with students.
- Martino Sorbaro: use simple analogy to explain staged-committed areas
- André Palóczy - Yes, need to keep practicing clearer ways of outlining general concepts on the board and be less wordy.
- Anne Glaser: Refrain from using the word ‘just’ and other words that are habits to use that may make something sound simpler than it really is.
- Billy Broderick: When describing the steps I’m taking, spell out uncommon things and, more importantly, explain why these are the steps we’re taking (e.g., “we’re using nano to open this file, but we could use any text editor, the important thing is that we’re creating and adding content to the file”)
- Dima Lituiev: Agree. As an action items, I can compare pandas table to excel spreadsheet, explain what head command does.
- Craig Boman - Going more slowly through demos.
- Juliana Velez Lujan
- Julia Piaskowski: Yes, and I *really* like the suggestion to use people’s research data for illustrating different data types to non-programmers.
- Nohemi Huanca-Nunez
- Saranya Canchi - I would lay out the benefits of piping as a way to introduce the problem and the commands first before delving into the lesson.
- Eva Sanchez: I feel great and I’m grateful for the feedback. I should focus more on why is it important to learn how to use GitHub.
-

Key Points

- 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.

IX. Wrap-Up and Homework for Tomorrow

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

To prepare for tomorrow, please:

1. Read about the two types of Carpentry workshops: self-organized (http://www.datacarpentry.org/self-organized-workshops/) and centrally-organized (http://www.datacarpentry.org/workshops-host/) and the checklists these pages link to.
2. Prepare for the live coding exercises. If you haven’t already, pick an episode from an existing Software or Data Carpentry lesson and read through it carefully. Tomorrow, you will use this to practice live coding for 3 minutes in groups of three. Your group members will comment on the delivery and content. Recommended episodes are listed here: https://carpentries.github.io/instructor-training/12-homework/.
3. **Create a Faded Example from a Lesson (10 min)**
   The following exercise should be done in groups of 2-3.
    2. Pick a block of code from an existing Software or Data Carpentry lesson, or from another lesson you have taught recently.
    2. Replace 2-3 pieces of the code with a blank.
    2. Write a question to test the student’s ability to correctly fill in that blank.
    2. Paste your faded example in the Google Doc.

Feedback (5 min)
The Trainer(s) will ask for feedback on the day in some form.

https://docs.google.com/forms/d/1TZjY9iFooFGYrVY8V4VFXBUj3yZzXDlWyUMUabS1ifM

Reflecting on the Day (10 min)
Before we wrap up for the day, take 5 minutes to think over everything we covered today. On a piece of paper, write down something that captures what you want to remember about the day. The Trainers won't look at this - it's just for you.
If you don't 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

# Day 2

X. Welcome Back

Questions (5-10 min)
Yesterday we asked you to read some resources about the logistics of teaching and running Carpentry workshops. Please add your questions about logistics and preparation to the Google Doc. We will answer some of these questions now and make sure to answer the rest by the end of the day.

Questions about Carpentry Workshop Format

- Katie Coburn
- Mara Sedlins: How can I find out if there are other Carpentry instructors in my area who may want to co-teach a workshop?
- Jacob Deppen - If I self-organize a workshop with myself and a non-Carpentries-trained partner, can I still call it a Carpentries workshop?
- Adriana Signorini
- Elizabeth Salmon: Can we offer shorter sessions, like for example a stand alone 90 minute workshop on OpenRefine?
- Matthias Bussonnier
- Jeffrey Weekley
- Sarvani Chadalapaka: Can we offer shorter but distributed sessions across 3-days kind of format? That way, I don't have to be away from work all day
- Chris Olson
- Martino Sorbaro Are we free to use carpentries teaching materials in other contexts, or in lessons organised by us (without claiming it to be an official workshop)?
- André Palóczy: What are The Carpentries' criteria/periodicity for doing centrally-organized workshops?
- Anne Glaser
- Billy Broderick: Is the only difference between the self- and centrally-organized workshops the assistance of Data/Software Carpentry staff for logistics and administrative issues?

- Dima Lituiev
- Craig Boman
- Juliana Velez Lujan
- Julia Piaskowski: I'd to deliver a more targeted training to my colleagues (ag scientists) - do I do this outside the Carpentries framework (similar question to Martino Sorbaro's)? Related: what does the process of developing new Carpentries content look like?
- Nohemi Huanca-Nunez
- Saranya Canchi

Faded Example of Coding Challenge

- Katie Coburn
  - human_age <- cats$age * ____
  - human_age <- as.factor(______)
- Mara Sedlins
  (from https://datacarpentry.org/r-socialsci/02-starting-with-data/)

  ```
  library(tidyverse)
  interviews <- read_csv("data/SAFI_clean.csv", __________)
  interviews
  ```

  If the missing data were encoded as "-999" instead of "NULL", how would you convert the missing values to "na" when importing the dataset? How do you know your code worked?

- Jacob Deppen

```
# find the sum of weights for each sex
surveys_df.groupby(['sex'])['weight'].sum()

# find the mean hindfoot_length recorded each year
surveys_df.______(['year'])['hindfoot_length'].______()

# find the number of unique species in each plot_id
surveys_df.______([______])['species_id'].______()

# find the total weight recorded each year
surveys_df.______([______])[______].______()
```

- Adriana Signorini

- Elizabeth Salmon: from Starting with Data: ecology
  - `# Count the number of samples by species`
  - `species_counts = surveys_df.groupby('______')['record_id'].______()`
    `________(species_counts)`
- Matthias Bussonnier
  - > Write a loop that calculates the same result as 5 ** 3 using multiplication (and without exponentiation).
  -
  - `result = 1`

```
for i in _____:
    result = result * 5
print(result)
```

  -
  - A) range(1,3)   # == 1,2,3 range inclusive
  - B) range(3)   # correct, start at 0, 3 inclusive
  - C) range(2)   # 0,1,2 - start at 0 but inclusive
  - D) range(1,2,3) # confuse range and list
  -

- Jeffrey Weekley
  - #Faded code example
    mystring = None
  - myfloat = None
  - myint = None
  -
  - # running code
  - if mystring == "hello":
  -     print("String: %?” % mystring)
  - if isinstance(myfloat, float) and myfloat == 10.0:
  -     print("Float: %?” % myfloat)
  - if isinstance(myint, int) and myint == 20:
  -     print("Integer: %?” % myint)

- Sarvani Chadalapaka

```
#Square matrix multiplication (Assume that the two matrices mat1 and mat2
can be multiplied. The result is stored in res matrix)
void multiply(int mat1[][N],
              int mat2[][N],
              int res[][N])
{
    int i, j, k;
    for (i = ?;  ?; ?)
    {
        for (j = ?; ?;?)
        {
            res[i][j] = 0;
            for (k = 0; k<N; k++)
               # print out the result matrix
        }
    }
}
```

- Chris Olson
  - SELECT * FROM person ORDER BY id DESC;
  - SELECT _____, person, quant FROM Survey ORDER BY taken ASC, ____ DESC;
  - SELECT DISTINCT quant, person _____ Survey _____ quant ASC;
- Martino Sorbaro (sorry, a bit too repetitive, not very interesting)
  - Example: if we have a new file called `data.txt` and we want to commit it,

    -
    - git add data.txt
    - git commit -m "adds the data file"
    -
    - suppose `git status` tells you there's a file already staged for commit, called `newfile.txt`. Complete the command:
    -
    - git commit ...?
    -
    - what happens if we don't write anything after git commit?
    - Let's now suppose we want to create an empty file and commit it. Complete the commands below.
    -
    - touch _________
    - git ________ emptyfile.txt
    - git ________ -m "created a new file"
- André Palóczy
    - animals = ["cats", "corals", "lemurs", "snakes"]
    - vehicles = ["car", "ship", "bicycle", "hovercraft"]
    - print("There are too many " + animals[0] + " in my " + vehicles[0])
    - >>> There are too many cats in my car
    -
    - **Q:** Fill in the missing code. Both loops are supposed to produce the same output.
    - for i in range(___(animals)):
    -         print("There are too many " + animals[i] + " in my " + vehicles[i])
    -
    - for animal, vehicle in ___(animals, vehicles):
    -   print("There are too many " + ___ + " in my " + ___)
-
- Anne Glaser
    - Problem: Remove the 4th column
    - `            coat weight likes_string age`
    - `1          calico    2.1            1   2`
    - `2           black    5.0            0   3`
    - `3           tabby    3.2            1   5`
    - `5 tortoiseshell    3.3            1   9`
    -
    - Cats[_______]
    -
    - A: [, -4]
    - B: (*, -1)
    - C: [, -1]
    - D: (*, 4)

- Billy Broderick: edit the existing file ~mars.txt~, compare it to earlier version, and commit it to git

```
nano mars.txt
# add "But the Mummy will appreciate the lack of humidity" to file
git __
git __ mars.txt
git commit __
```

- Dima Lituiev
  You are given a table (`species`) with following fields:
  ```
  Field  Data Type     Motivation    Table(s)
  genus  TEXT   Field contains text data   species
  hindfoot_length      REAL   Field contains measured numeric data    surveys
  sex    TEXT   Field contains text data   surveys
  species       TEXT   Field contains text data   species
  taxa   TEXT   Field contains text data   species
  weight REAL   Field contains measured numerical data in gram surveys
  year   INTEGER       Allows for meaningful arithmetic and comparisons
  surveys
  month  INTEGER       Having data as numeric allows for meaningful arithmetic and
  comparisons   surveys
  day    INTEGER       Having data as numeric allows for meaningful arithmetic and
  comparisons   surveys
  ```

  Complete a query that extracts full rows of the table that have weight under 1kg seen anytime before 2018.
  ```
  SELECT ___ FROM species
  WHERE ___ AND ___;
  ```

  -
- Craig Boman
  - SELECT * FROM _____ AS t
  - SELECT * FROM table WHERE __.column
  - SELECT * FROM table WHERE _.column < __
- Juliana Velez Lujan
- Julia Piaskowski
  - # extract columns that are factors from data "warpbreaks" and assign to new data frame
  -
  - data(warpbreaks)
  - str(_?__)
  - wb2 <- warpbreaks[__?__]
  -
  - # use lapply for indexing
  -
  - index <- apply(warpbreaks, 2, __?__)
  - wb2 <- warpbreaks[__?___]
  -
- Nohemi Huanca-Nunez
- Saranya Canchi
  - # Get the dimensions of a data frame, view the contents and subset the 1st and 3rd columns to a new data frame
  - ---(gapminder)
  - ----(gapminder)
  - Gapminder_subset <- gapminder-------
  - Alternatively
  - Gapminder_subset <- -------(gapminder--------)

Key Points

- 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.

XII. Building Teaching Skill: Live Coding

https://carpentries.github.io/instructor-training/14-live/index.html

Why Live Coding?

https://youtu.be/bXxBeNkKmJE

Up and Down (5 min)
List some advantages and challenges of live coding from both a learner's and an instructor's point of view in the Google Doc.

The Bad and the Good (15 min)
Watch this video of live coding done poorly and this video of live coding done right as a group and then summarize your feedback on both in the Google Doc. 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.

Content +
Content -
Presentation +
Presentation -

- Katie Coburn

Content +

- Presumably informative

Content -

- difficult to understand the content very well since the presentation was so bad

Presentation + not much

-

Presentation - hard to read, background website, notifications on his laptop, doesn't speak loudly, barely looks at his audience,

- Mara Sedlins
  - Content +

        - gives multiple examples of for loops doing slightly different things so that learners can see the common pattern
    - Content -
        - example is pretty abstract/generic, not tied to a real research example
    - Presentation +
        - verbally explains what the code is doing after typing and running it
    - Presentation -
        - could have taken a moment to explain the error message he got
- Jacob Deppen
    - Content +
        - Repeated the same thing with different variable names to show which parts were necessary
    - Content -
        - Did not really explain why this was useful
        - Could have used his error as a place to teach something
    - Presentation +
        - Does some explanation as he was typing
    - Presentation -
        - Small thing, but he was typing at the bottom of the screen where it can be tough for people in the back to see

- Adriana Signorini
    - Presentation - more projection and layout is not accessible to everyone- the mind process should be shared.
    -
- Elizabeth Salmon
    Content +
    Content -
    Presentation +
    Presentation - would like instructor to speak out loud while coding; more interaction with audience, heads in the way of screen, colors may have low visibility for some
- Matthias Bussonnier
    - First
        - Font too small,
        - Dark terminal.
        - Notification.
        - Prompt too complicated.
        - Does not explain wht he writes.
        - Is sitting, lack dynamism.
        - Not asking for feedbck.
        - Not looking at the audiance.
        - Goes too low onthe screen.
    - Second
        -
    -
- Jeffrey Weekley
    - Presenter was not seeking any feedback from the audience. He wasn't even looking at them. It was obvious that no one was following along.
    - His window was too low, so the text was blocked.
- Sarvani Chadalapaka

  - Font too small
  - Made correction without really explaining why
  - A little too fast but not too fast if you're familiar with the subject
  - Whatsapp notifications (intended?)
- Chris Olson
  - Content +
    - Ran the code with variations to illustrate how different variables can be used
  - Content -
    - Could have explained the practicality of the example - why would we do this?
  - Presentation +
    - Went at a slow pace and explained everything he did
  - Presentation -
    - Small font
- Martino Sorbaro
  - Content +
    - Explains flexibility of variable name. Good for those who have no good understanding of maths.
  - Content -
    - Does not explain what his mistake was when he got an error.
  - Presentation +
    - Shows how it's actually done. No damn powerpoints.
  - Presentation -
    - Does not explain while he is writing, giving the impression of long periods of wasted time
    - Looks at the screen while talking even when not coding. Looks like he is reading and does not engage with the audience. He is not natural.
- André Palóczy
  - Content +
    - Good examples
  - Content -
  - Presentation +
    - Well-highlighted and large font
  - Presentation -
    - Whatsapping while teaching
    - GPS monotonic voice
    - Never looked at the audience (and missed a struggling learner (with a red minute card))
    - Some parts were suddenly too fast
- Anne Glaser
  - Content + Took time walking through the examples and didn't seemed rushed
  - Content - Didn't stop to explain what he was doing or give background. Didn't engage the students, looked at screen entire time.
  - Presentation + Stepped through each example line by line
  - Presentation - Text was a bit small and hard to read
- Billy Broderick
  - Content +: explained how variable assignment works, that the name is arbitrary, and demonstrated this
  - Content -: didn't explain the syntax of for loops or the head command at all
  - Presentation +: went slowly, made a mistake and then corrected it

    - Presentation -: didn’t talk through and explain while typing, didn’t explain what the typo was. Looking at screen the whole time, didn’t look up at the learners
- Dima Lituiev:
    - Presentation + : good color scheme
    - Content +: good explanation of variables
- Craig Boman

    Content +
    - Good content

    Content -
    - Could have explained more clearly what for loops are
    - Explain errors

    Presentation +
    - Spoke clearly and loudly

    Presentation -
    - Make sure to explain large blocks of text you are typing
    - Interact with the audience more
    - Ignores sticky note
    - Turn off social notifications during presentation
    - Text editor was using less accessible font colors
- Juliana Velez Lujan
- Julia Piaskowski
    - Content + seemed good (we lack some context to evaluate this), he explained what he was doing while he did it and why things like the cursor changed
    - Content - did not cover much,
    - Presentation + also seemed fine
    - Presentation - (font seemed small, but maybe not for the people there), it’s okay to sit, but he should probably look up more at his audience
- Nohemi Huanca-Nunez
- Saranya Canchi
    - Content + -> Did good job explaining the loop and flexibility of variable name
    - Content - -> Could have also shown the use of tab autocomplete and history. Also did not explain the error due to typo
    - Presentation + -> Good pace, clarity in communication
    - Presentation - -> Seemed to be engrossed in the laptop , not looking at the participants to make any eye contact
-

Second video:

https://www.youtube.com/watch?v=SkPmwe_WjeY&feature=youtu.be

Feedback

Content +
Content -
Presentation +
Presentation -

- Katie Coburn
    - Content + good, includes a typo which is useful since those happen a lot
    - Content - good
    - Presentation + Says commands out loud as he types them, appears more engaged with audience, reaches to screen to explain content, thoroughly explains each command. Font bigger, screen clearer, no interruptions, his screen looks more similar to his learners'
    - Presentation - Very good!
- Mara Sedlins
    - Content +
        - gives more examples, changing the variable name several times so that learners can see the common patterns and build their mental models of how a for loop works
    - Content -
        - example is still pretty abstract/generic, not tied to a real research example
    - Presentation +
        - does more verbal explanation of what he does as he is doing it; explains the error message in detail as it occurs
    - Presentation -
        - pace may be too slow for some learners
- Jacob Deppen
    - Content +
        - Did a nice job going back and stepping through the code that he had just written
    - Content -
        - Didn't need to discourage the use of semi-colon
    - Presentation +
        - Positioned the terminal window so that it could be seen by everyone
    - Presentation -
        - Still could have paused to make sure folks were following him on their laptops
- Adriana Signorini
    - Presentation + professor stops and explains step by step - bigger text - identifies mistake and corrects it on the spot - speaks up his thinking process- engages with the audience - slower pace! Professor body language is positive; he seems happier- dynamic lecture -
- Elizabeth Salmon
    - Content + provided different examples/options for same concept, explained error
    - Content -
    - Presentation + good pacing, good explanations, yay talking out loud while coding, acknowledged that this takes time and may still make errors, engaged with audience
    - Presentation -
- Matthias Bussonnier
    - White background, black font, Zoomed in.
    - No notification.
    - Simple $PS1
    - Standing,
    - Getting feedback from the audience.
        - Explain what he will do,
        - Does it
        - Explain what he did step by step.
    - Could have use indentation to make loop more visible.
    -
- Jeffrey Weekley
    - His setup was better and more readable.

    - He is going slower
    - Looking at the audience for feedback
    - Pointing with hands and reviewing the material
- Sarvani Chadalapaka
    - Explains why he does what he does
    - The terminal is at a good height
    - Explains the naming convention- good
    - Fixes the error AFTER explaining the error
    - Also, tells them the tricks of the trade (like up arrow and semicolon)
-
- Chris Olson
    - Content +
        - Went into more detail on things
        - Made a mistake and explained what happened and his thought process
    - Content -
    - Presentation +
        - Standing up seemed a bit more engaging to me
    - Presentation -
        - I find it harder to read the white background (but maybe that's just me)
- Martino Sorbaro
    - Content +
        - The explanation is much more in-depth.
        - Uses the error to explain something more.
    - Content -
        - Probably not needed to give a third example with a third variable name. Otherwise great.
    - Presentation +
        - Stands, looks at the audience, uses gestures, his tone is not flat and uniform.
    - Presentation -
        - It takes much longer.
- André Palóczy
    - Content +:
        - Explaining what bash is doing under the hood step by step while going through the for loop and `head`
        - Spelled out what is happening on each iteration of the for loop
    - Content -:
    - Presentation +:
        - No whatsapp
        - More dynamic, moves and points to commands on the screen
        - Engaged with the class, attentive to red sticky notes
        - Looks at the class
        - No more GPS robot voice
        - Explained the error when made a typo (`headn` instead of `head`)
        - "Trust me, I've been there". Learners can relate, fosters growth-oriented mindset
    - Presentation -
        - The colors of the syntax highlighting are gone
- Anne Glaser
    - Content + Took time to go back through the logic to explain how things happened.
    - Content -
    - Presentation + Screen was clear to read. Nothing else on the screen. Standing & Engaged the audience just by gesturing and looking at them.

    - Presentation -
- Billy Broderick
    - Content +: explains concepts as they come up, instead of after the command runs. Explains what semicolons are after hitting the up arrow key
    - Content -: didn't explain head syntax
    - Presentation +: engaged, looking at audience, pointing things out on the screen. Reads what he's typing out loud
    - Presentation -:
- Dima Lituiev
    - Content +: good explanation of variables, command output, command line interface (including newline and semicolons), errors
- Craig Boman
    - Content +
        - Content was clear; very good
    - Content -
    - Presentation + :
        - Acknowledged sticky notes
        - wasn't buried in laptop
        - Standing at podium worked better than sitting
        - Used more accessible text editor
        - Interacts more with projected image to be more clearly what he is referencing
    - Presentation -
- Juliana Velez Lujan
- Julia Piaskowski
    - Content + very good - he told us from the start what we would be learning. I now know how to do loop in bash
    - Content -
    - Presentation + better, large font, he stopped and explained what was going on, good use of mistakes, he looked up more at his audience
    - Presentation -
- Nohemi Huanca-Nunez
- Saranya Canchi
    - Content + - Clear explanation and reasoning for the actions. The commands were clearly explained including errors. He also connected what they had learned earlier regarding the up arrow key and showed how it could be used here.
    - Content - -> Could have explained the structure of the loop a bit more for example what is variable name and why should we use it
    - Presentation + - Good pace, and eye contact with the audience. Took the time to explain line by line the code and the results.
    - Presentation -

Feedback:

Sticky Notes

Practice Teaching (20 min)

Teach 3 minutes of your chosen lesson episode using live coding to one or two fellow trainees, then swap and watch while the other person(s) live codes for you. Explain in advance to your fellow trainee(s) what you will be teaching and what the learners you teach it to are expected to be familiar with. **Don't record the live coding sessions.** Give each other feedback using the 2x2 rubric we discussed previously and enter the feedback you received in the Google Doc. To make this exercise as similar to the workshop experience as possible, ask your fellow trainees to code along with you, as if they were learners at your workshop.

Content +
Content -
Presentation +
Presentation -

- Katie Coburn
  Content +: overview of the material that students would have learned in previous lessons, focused on the elements that were of most interest
  Content -: a little hard to follow along with since not everyone had the software installed
  Presentation +: good pace, good explanations of things
  Presentation -: create drawings somehow to clarify
- Mara Sedlins:
  - Content +: worked directly from the Data Carpentry content, no specific feedback on this
  - Content -: would want to give more context about what tidyverse is, how it's different from base R;
  - Presentation +: did a good job of saying what I was doing as I was typing the code
  - Presentation -: for the read_csv command, it's a little confusing that the directory location was "data/filename.csv" but the file explorer pane in RStudio was displaying a different directory -- be clearer about the difference between that and what the working directory is
- Jacob Deppen
  - Content + good explanation of types
  - Content - could have introduced topic more thoroughly
- Adriana Signorini
  - Mentioning the outcomes for the session
  - Eliciting info from the class
  - Easy examples to follow
- Elizabeth Salmon
  - Content + provided learning outcomes in the beginning, engaging examples
  - Content - hard to troubleshoot mistakes
  - Presentation + interactive
  - Presentation - make use of whiteboard as well
- Matthias Bussonnier
- Jeffrey Weekley
  - Content + Lesson goal was presented at the top (setting expectations)
  - Content - Example was simple and easy to follow. Taught a useful skill
  - Presentation + Followed best practices
- Sarvani Chadalapaka

- Chris Olson
  - Content +: Good pace
  - Content -: Didn't give the background as to why, what a repo is, what a commit is,
  - Presentation +: Leaving the laptop and pointing at the screen and explaining results helpful
  - Presentation -: Don't use words you haven't defined yet, slack alerts were distracting
- Martino Sorbaro
  - Content +: Choice of content (introductory git)
  - Content -: Explain why -m is needed

    - Presentation +: Took time to explain details.
    - Presentation -: Rushed at the end.
- André Palóczy
    - Pres -: Didn't explain that indentation in the for loop matters in python (no `end` statement)
    - Pres +-: Good/bad color scheme (IPython's defaul) was good for some, bad for others.
- Anne Glaser
    - Content + Made a mistake and explained how to correct it and used the opportunity to use the up arrow and explain it.
    - Content - Should have explained the combine function
    - Presentation +
    - Presentation -
- Billy Broderick:
    - +: went slowly, explained what I was going to do before I started
    - -: switching between terminal and text editor could be confusing for people, so take more time explaining what's happening there. When looking at git outputs, walk through them line-by-line instead of just giving the overview. Also, if more time, would be helpful to have an overview diagram of git to explain basic concepts
- Dima Lituiev:
  (+) rephrasing
  (+) meaningful variable names
  (-) no check in with students
- Craig Boman
    - + went slowly
    - + spoke clearly
    - - more goal setting
    - - explain more clearly formatting convention for sql commands versus variables
- Juliana Velez Lujan
- Julia Piaskowski: I had a few technical issues, but Jacob indicated that I clearly explained aspects of how to declare a variable in C, what a header file is and some other rules of C. Some more idiosyncratic rules were not explained adequately. More elaboration would have helped resolve that. I also should have said why a person would want to learn this (why is it useful).
- Nohemi Huanca-Nunezu
- Saranya Canchi - I was able to explain clearly the commands used but should have explained clearly the difference between script pane and console pane. Also should have showed the data set as a table for view and given some more background about the dataset.

Feedback

Key Points

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

XI. Building Teaching Skill: Lesson Study

https://carpentries.github.io/instructor-training/15-lesson-study/index.html

Learner Profiles

Learner Profiles (10 min)
Read Software Carpentry's learner profiles and then write one that describes a fictional colleague of your own. Who are they, what problems do they face, and how will this training help them? Be as specific as possible.
Enter your learner profile into the Google Doc.

https://software-carpentry.org/audience/

- Katie Coburn
    - Aidan Annotate
    -
    - Dr. Aidan Annotate is a postdoctoral fellow in computational chemistry at the City University of New York (CUNY). He has learned programming by experience as a graduate student and knows both R and Python, among others.
    -
    - Dr. Annotate and his colleagues work on developing tools to support organic synthesis. He is creating an R package that he plans to distribute through the Comprehensive R Archive Network (CRAN). Although Dr. Annotate is fluent in R and writes effective functions, he claims he never needs to annotate his code. After all, he is the only one who needs to read it and understand it! (He certainly never forgets what he was doing and gets confused!)
    -
    - Software Carpentry can teach Dr. Annotate the virtues of version control and Git, along with the importance of good commenting practices and data management – not just for consumers but for Dr. Annotate himself.
- Mara Sedlins
    - Maria is a graduate student in literature who is interested in computational approaches to text analysis and has taken several Python tutorials online. She would like to use digital humanities methodologies to generate interesting new questions about Shakespeare's plays and supplement her literary interpretations. She has access to a digital copy of Shakespeare's entire corpus of text, but needs help understanding what she can do with it in Python. She is interested in the relative frequencies of particular words, as well as which words co-occur, and she would like to look for trends across plays written at different point in time. A Python workshop will help orient her to producing answers to her questions and creating/interpreting visualizations of the data she produces. Maria is hearing impaired and prefers that presenters use a microphone.
- Jacob Deppen

    - Ann Thropology - Ann is a junior faculty biological anthropologist who studies skeletal morphology of human ancestors. She does a lot of traditional statistics and typically uses SPSS because she was introduced to it in an undergraduate Psychology course. Most of her data entry and simple visualization is done with Excel. She has heard that a lot of people have started to use scripting languages like R and/or Python for analysis and wants to see what it might have to offer her. She installed RStudio, but doesn't know what she should do next. The Carpentries training would help Ann identify places in her typical workflow that could be improved by writing code and give her a base level of confidence to pursue further training. Ann has a visual impairment so she usually sits in the front of the class.
- Adriana Signorini
    - Billy Cheng is a lecturer teaching introduction to computer science. Billy teaches a large class. He is new in the discipline and this workshop could help him with his teaching.
      How? Going through the practice, teaching demos and receiving feedback. Intro to how learning works. Interacting with others.
    - why, how, when
- Elizabeth Salmon

    Dan is an employee at the UC Merced Library. He earned a BA in English in 2005 and prior to coming to UC Merced he managing shipping logistics for his family's food importing and distribution business. He is considering pursuing an MBA.

    Dan is regularly tasked with compiling and analyzing library statistics, creating graphs and charts, and drafting reports. Typically, he produces the similar reports each. This data tends to come from multiple sources and requires cleaning and standardization before he can start analyzing it. Dan currently uses excel and spends a lot of time cleaning his data. Dan has also been tasked with running the summer inventory project this year.

    Dan is color blind.

    Library Carpentry will teach Dan how to use OpenRefine to clean and standardize his data by using the facet and clustering tools; and how to more effectively accomplish these tasks in the future.

- Matthias Bussonnier
- Jeffrey Weekley
    - Daniela Norte-Adueño is a Master's student in the computational biology group at the University of California. She is beginning her studies of the DNA of echinoderms. In particular, she will be working on studying Pisaster Ochraceus and Sea Star Wasting Syndrome. She has dual a bachelor's of science in biology

    and organic chemistry, but is new to computational biology and the usual tools used in comparative analysis of DNA. She has strong math skills, but doesn't know much coding.

    - Because Daniela grew up far from the ocean, she was always fascinated by her visits with her family to the beach and dreamed of being a marine biologist. Her parents immigrated to the US from Latin America (central Mexico) and she is not only the first person to attend and graduate from university, no one in her family or her family's wide circle of friends has a child pursuing a postgraduate degree. She sometimes struggles with finding a role model. Eventually, she'd like to get her PhD.

    - Her professor is coaching her to be a better scuba diver, and how to collect samples in the field, but she isn't able to spend the extra time coaching her on coding (just higher-level methodologies). Still, she is very eager to gain these skills on her own.

- Sarvani Chadalapaka:
- Chris Olson: Vanessa is a second year graduate student in Biological Oceanography. She has no formal programming training, but has worked with several small datasets in undergraduate classes. Vanessa's advisor has just given her 10 years of fish samples data in monthly csv files and asked her to make a map showing a certain species over time. Software carpentries will teach her how to clean and normalized this hand collected data, and write python scripts to extract and plot the relevant data.
- Martino Sorbaro:
    - Jim is a master's student in epidemiology and biostatistics. His course includes an introduction to R, but this is taught very quickly, and he hasn't had any previous exposure to coding. Therefore, Jim is anxious about the possibility of failing the test, and has so far been struggling in finding a way to teach himself R, which caused him to feel stressed and unmotivated about coding.
    - The particular set of skills he needs revolves around reading data, computing means and standard deviations, performing simple statistical tests to compare samples and their statistics. Some plotting would also be helpful. He has had some classes about STATA before.
    - A carpentry workshop would help Jim with a more gentle introduction to coding, so that he will feel less frustrated and demotivated by the process. It would also give him the practical notions needed to start performing the tasks required, but also enable him to know how to learn new concepts by himself, and troubleshoot his errors by himself.
- André Palóczy:
    - Bob Smith received a BS and a MS in Public Policy from the University of Washington in 2005 and 2007, respectively, and he is only familiar with MS Office and Windows XP. He then took a gap year, during which he worked for a data

science company analyzing commuter data for the Greater Seattle bus and subway public transportation network. He was both the head of and the only programmer in the company’s data science department before the company went out of business in 2009.

    -
    - Bob analyzed commuter boarding and dropoff locations, times of day in Excel spreadsheets, plotted line, bar and pie charts in png and describes the results in the annual report his company delivers to the Seattle Public Transportation Bureau. He keeps all entries in a Notepad text file and often has to copy-paste subsets of the data based on month, time of day, or sub-network.
    -
    - Bob has impaired hearing and uses a hearing aid on his right ear.
    -
    - SWC will help Bob be more productive by teaching him how to analyze and visualize tabular data with Python and pandas.
- Anne Glaser
    - Beth A is a Product Manager whose background is in Plant Science. She has some background using matlab and R, but as a product manager is now interested in learning java to better understand and test her application's codebase. Her daily job has her working daily with researchers and developers. She is able to understand the researcher's needs, but need to be able to communicate those needs to the developers and wants more background on various coding languages. She is learning to learn jira and git as well.
- Billy Broderick
    - Roman is a neuroscience PhD student. He has taken one course in grad school that used Matlab as a way of solving simple regression and statistics problem, but has never taken a dedicated programming course. For his research, he gives mice a drug and then records the activity of hippocampal neurons while they attempt to navigate a maze, comparing the neural activity and the mice’s performance to mice given a placebo. He inherited a Matlab script that pre-processes the data and extracts when neurons were spiking, which he copies and changes the parameters (set at the top of the file) for each recording session or mouse. He then copies the output of this script into an Excel spreadsheet in order to visualize it. Software Carpentry will help him understand how to write functions so he only needs one version of the Matlab script, create plots to understand the output, and write scripts that connect these steps.
- Dima Lituiev

    Anna is a PhD student in Molecular Biosciences. She has taken introductory course to R during her undergraduate studies and Bioengineering course, which used Matlab. She used R two years ago to produce box-plots from her data for a paper. With her current project, she obtained RNA-seq data from sequencing facility, which contains gene names and their expression levels. She wants to learn how to import, combine, transform, and analyse her data using R in order to compare her samples in a repeatable way, so that she can run same analysis again on her next experiment.

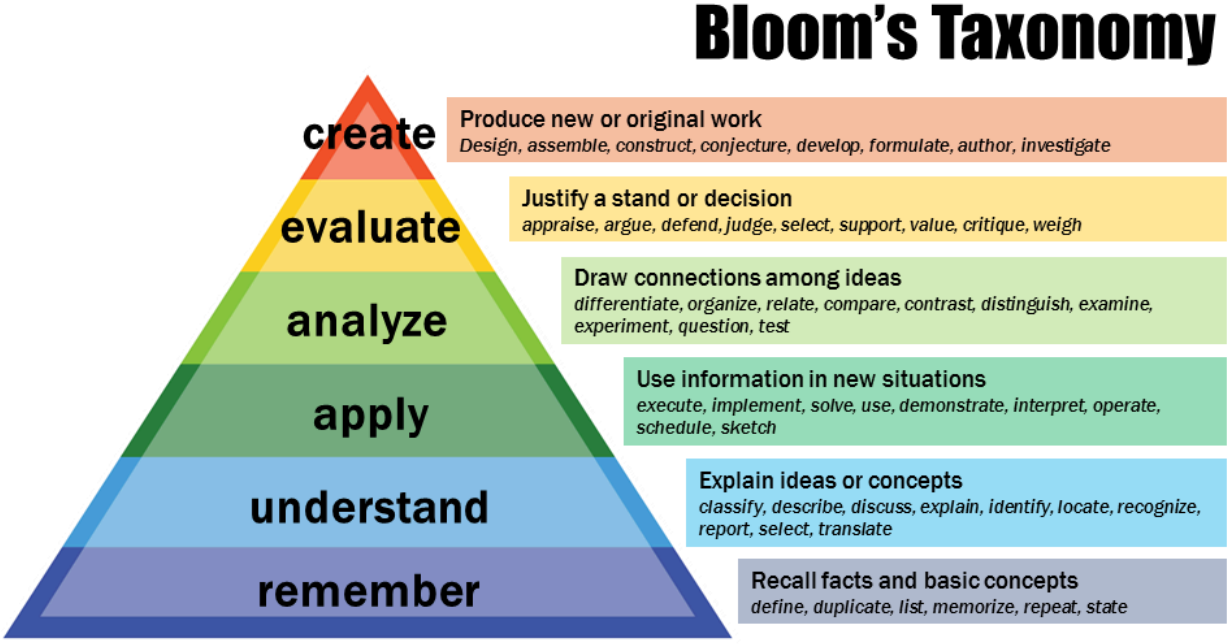
Anna suffers from undiagnosed anxiety and was diagnosed high blood pressure since five years.

SWC will refresh Anna's skills in data manipulation in R, which will help her to prepare her data and gain preliminaries for an advanced statistics workshop she's taking next semester.

- Craig Boman
    - Barbara Booker is a cataloger at a library with a Master's in Library Science. She has spent the last few decades cataloging books and is confident with Excel. She is awesome at describing books quickly or finding metadata that already exists. But she is aware that there may be a quicker way to find metadata about books more quickly or in a batch. She is interested in learning more about how she might batch query her metadata vendor for a list of book ISBNs rather than having to look for information one book at a time.
- Juliana Velez Lujan
- Julia Piaskowski:
    - Laura is a senior research who manages all the data on a large multi-institutional research project. She has a PhD in bioinformatics, is very knowledgeable in her domain subject matter and is comfortable working on the command line. She manages all the project data on her work desktop with the password to that hidden in her lab. She keeps a folder of current files, their accompanying "README.txt" files, and old versions in a "old files" directory. Every time someone needs the updated data, she emails the relevant files and saves a copy of that email in the aforementioned project data directory. A Software carpentry workshop on version control and secure cloud storage could help Laura manage that data more efficiently and securely.
- Nohemi Huanca-Nunez
- Saranya Canchi - Baron Von Brain is a Ph.D student in Neurosciences who is using multi panel electrical recordings to understand the visual circuit. He completed his B.Sc in biology. He took a introduction to programming in Python during his senior year and has not had very extensive programing experience. He currently has continuous recordings from over 5000 rods that are taken over two times a day for 7 days. His goal is to understand the changes in the signal intensity with time and space. He currently is using manual tracking to track changes but needs a better way to manage the data and get the required output. He suffers from seasonal depression. SWC can teach him the tools to manage his large data, write code to automate the parsing of data for analysis and teach him about version control to keep track of changes with experiments.
- Eva Sanchez: Marian is a Lecturer at a Community College with a Biology background. She would like to teach R to freshman and show them different applications for such programming language. She will use the environment called R studio, and will start with basic statistics that will help students with their first year research projects. She will continue her teaching by developing plots and beautiful figures to be added to lab

reports/research proposals. If Marian decides to take a software carpentry workshop, she will be able to enhance her teaching skills, as well as learn other tools to manage data and continue helping students.

Working With Learning Objectives


Bloom's Taxonomy
create
Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate
evaluate
Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critique, weigh
analyze
Draw connections among ideas
differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test
apply
Use information in new situations
execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch
understand
Explain ideas or concepts
classify, describe, discuss, explain, identify, locate, recognize, report, select, translate
remember
Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state

Vanderbilt University Center for Teaching

Evaluate SWC and DC Learning Objectives (10 min)
Take a minute to select one learning objective from one of the Carpentry lessons, then complete the following steps to evaluate it.

1. Identify the learning objective verb. How specifically does this verb describe the desired learner outcome?
2. In your opinion, does the lesson do an effective job of meeting the stated objective?
3. Does the lesson meet any objectives that are not stated in the objectives section?

Instructor Notes

https://swcarpentry.github.io/r-novice-gapminder/guide/

Designing Good Challenges

Challenge 4
Given the following code:

```
N
```

```
m <- matrix(1:18, nrow=3, ncol=6)
m
      [,1] [,2] [,3] [,4]  [,5]  [,6]
 [1,]  1      4      7   10   13   16
 [2,]  2      5      8   11   14   17
 [3,]  3      6      9   12   15   18
```

Which of the following commands will extract the values 11 and 14?

A. m[2,4,2,5]
B. m[2:5]
C. m[4:5,2]
D. m[2,c(4,5)]

What Can You Learn? (10 min)
For your chosen lesson, pick one of the existing exercises, or use the faded example problems you wrote yesterday. Identify one (or more) mistakes that a learner could make doing that exercise that would provide valuable information to you as the instructor. What information would it provide and how would this affect your next steps in teaching the lesson?
Discuss your thoughts with a partner and add to the Google Doc.

Feedback On Your Challenges (15 min)
With these goals in mind, pair up with a partner to discuss the MCQ and faded example problems that you wrote yesterday. Give each other specific, actionable feedback that follows our 2x2 framework. Use that feedback to make at least one modification to your exercise(s). Discuss in the Google Doc the change you made and how it will help you get more useful information about your learners.

-
-
-

Key Points

- To teach effectively, you have to know *who* you are teaching.
- Good learning objectives communicate the intended effect of a lesson on its learners.
- A good exercise provides useful guidance to instructors about next steps needed in teaching.

## Mid-morning break

XIII. Building Teaching Skill: Performance Revised

https://carpentries.github.io/instructor-training/17-performance/index.html

Round Two (35 min)

Get back into the same groups you did your live coding with. Take turns re-teaching your chosen live coding session, making sure to incorporate changes based on the feedback you received. Give feedback to each other and add it to the Google Doc using our 2x2 rubric. Also add some feedback to the Google Doc on the following questions: 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?

Key Points

- (Reflective) Practice makes perfect.

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?

- Katie Coburn
  - Provided feedback this time on the content choice (C++ vs. R) and regarding when to stop for assessment. Did a good job of presenting the end goal at the beginning of the lesson. Demonstration of polling the audience for their understanding of the new ideas presented. Definitely a possible opportunity for drawing on a whiteboard (matrix multiplication).
- Mara Sedlins
  - I spent more time orienting the learner to what a working directory means in RStudio and pointed out that this is different than the file pane. I ended up going at a slower pace, which was helpful but didn't allow me to get to as much material. One piece of important information that I missed was explaining what the <- symbol does in R. If I taught it again, I would want to think carefully about how to orient people to the RStudio environment in general, and plan for what they need to understand even before reading in a data file.
- Jacob Deppen - added a clear use case at the start to try to connect the principles to something people have experienced; The lesson was a bit dry as I taught it, so I would like to add more of my experience. For example, explaining a time when checking a dtype right away saved me a lot of pain later
- Adriana Signorini-
  - Outcomes at the onset
  - Survey the class
  - Keep someone else's motivation in mind besides yourself.
  - Board graphs to supplement the info
  - Use of analogies
- Elizabeth Salmon
  - I feel like we were more critical (in a good way I hope!) with our last teaching example; stated the learning outcome in the introduction; importance of surveying the class for previous experience with coding language
- Matthias Bussonnier
- Jeffrey Weekley
  - The next example given was a lot more complex than the previous example, so we had to stop, survey the audience and make some analogies. Once we reviewed the basic principles, the novice audience members caught up again.
- Sarvani Chadalapaka
  - Content - Lesson goal needs to be more relatable (motivations)
  - Example with loops need to be more clear (maybe get a pseudo code) to explain the basic algorithm
- Chris Olson
  - Might be helpful to use analogies for git
  - Git has a lot of terminology that needs to be addressed
- Martino Sorbaro

    - Terminal was not ready for second session (git not clean), so I had to clean it up, and that could not be explained quickly.
    - I checked the students could see the pointer
    - I gave a little summary at the end
- André Palóczy
- Anne Glaser: Stopped to explain some basic concepts based on feedback from last time - combine. The arrow vs equal sign in R is a bit confusing and could've skipped over for the time being.
- Billy Broderick: incorporated most of previous feedback, which worked well. I forgot to delete the old directory and so made some live mistakes, having to go back again, which were effective. One new thing that came up is that I could've explained the output of ls better (why does this one have an extension and this other one doesn't). Also a difficulty I noticed is, because this is the second time teaching it in not too great a space of time, I wanted to go faster and so needed to pay extra attention to make myself slow down.
- Dima Lituiev: I would take more time for students' feedback
- Craig Boman
    - Good progress from first session
    - Make sure to mention goal of sql query
- Juliana Velez Lujan
- Julia Piaskowski: could not do it, too many technical problems on my end, could not troubleshoot them all in the time
- Nohemi Huanca-Nunez
- Saranya Canchi - Incorporated feedback from last time to show the dataset and tell some basic information about the dataset. I was also able to incorporate my personal preferences and reasons for looking at a dataset.
- Eva Sanchez. Intro to R studio. The second round I felt more confident and was able to talk, show a dataset, click on several sections and provide a better explanation regarding R studio. I didn't look at the audience directly, need to work on this.

This is the rubric that is used to evaluate new instructors to the Carpentries:
https://carpentries.github.io/instructor-training/demos_rubric/

Take a look and see how it maps on to your own experience with the teaching demos we have done in this workshop, and think about things you might be able to do to continue to improve and refine your content and delivery style.

Managing a Diverse Classroom

https://carpentries.github.io/instructor-training/18-management/index.html

What Are the Challenges? (5 min)

What are some of the challenges you might expect when teaching learners with a broad range of expertise? Discuss with a partner and put your thoughts in the Google Doc.

- Katie Coburn
    - When some students are novices and others are experts (and may know the material better than you!), you sometimes have to choose a level to teach to. Do you bore the experts? Do you challenge the novices? Do you find a midpoint and ask helpers to fill in the gaps?
    - This could also pose difficulties because you, as the instructor, want all the learners to feel that they "got something" out of the workshop. If you have experts in attendance, they likely will not feel that they "got anything" from a relatively basic lesson. However, novices will definitely not "get" anything

from a lesson that is way over their heads. So satisfying all learners at the end of the day could be difficult in a very diverse classroom environment. Maybe you could have a general goal that can be modified depending on level of expertise?

- Mara Sedlins
  - Learners may have different expectations about what they will be getting out of the workshop, different learning goals. There might be a few people who are consistently falling behind, while others may ask questions that demonstrate their expertise and might lead to tangents that won't be accessible or relevant to others in the room. Some advantages of this might be that learners with less experience could be exposed to more sophisticated examples of what the programming language can do - but it would be important to frame this as something that it's okay if they don't get it, and then transition back to the main curriculum in a way that frames the more advanced "tangent" question in relation to what I'm teaching.
- Jacob Deppen
  - It is really helpful to try to identify expertise early. That can give you context for understanding how to approach a question from that learner later.
  - It is essential to acknowledge the struggles of those that are struggling. Nothing is more frustrating than struggling in silence, so always providing an off-ramp to those who are struggling (with sticky notes, peer help, etc.)
- Adriana Signorini
  - Problems: you go too slow or too fast for some. If too fast, some will be lost for the rest of the time
  - Solutions: Pre-survey them to know their prior knowledge.
- Elizabeth Salmon
  - Pacing to keep all levels engaged; ensuring people leave with their questions answered esp. if they are at a higher or expert level; providing examples that relate to audience members own work or research; uneven participation in group activities; students falling behind may not be as willing to ask for help
- Matthias Bussonnier
- Jeffrey Weekley
  - With co-teaching, you can break the group into two smaller groups: one that can go through the material more quickly, and a second group that needs more coaching and one-on-one help. As the fast group finishes, they may be able to then do peer-mentoring with the folks in the slower group. This allows both groups to remain engaged in the activity, and even gives some of the advanced students a chance to learn some teaching skills.
  - Remain available during the breaks so that students who need some extra time can spend time with you one-on-one. Also, you might consider scheduling an hour (or so) before hand to help people get their environments set up.
- Sarvani Chadalapaka
  - Differences in expertise level and expectations
  - With co-instructor it will be easier to keep the session interesting for all skill levels
  - It certainly is important for attendees with more skill level to also take something back from the session - maybe identify a common goal that is challenging for both groups?
  - It is extremely important for new users to learn something useful as well, otherwise, they lose motivation to attend the session again
- Chris Olson
  - More advanced students may tune out and start talking if they are next to each other, distracting and deterring the more novice students. It could be really helpful to split up advanced people and put them next to more novice people and encourage them to work together. Also, generally changing who sits next to who can help.

    - It’s important for others to not get left behind but also keep the advanced people engaged. The true test of understanding something is being able to explain it, so maybe have the advanced users explain something.
- Martino Sorbaro
    - The problem is mostly about people lagging behind and people going ahead.
    - We need to keep a good communication making sure we understand who is in which situation (yellow/blue sticky notes method, helpers, diagnostic MCQs…)
    - Peer-learning can help. Students can work together to mitigate the problem
    - Identify problems that may be unrelated to prior knowledge: problems with their computer, disabilities...
- André Palóczy
    - When more than one learner has fallen behind, have ppl who are ahead help them if there are not enough helpers
    - Be mindful of very introverted people, try to not put anyone you feel might be like that on the spot
    - Have ppl with hearing/visual impairment sit closer to the screen/(sound sources? If the sound is not great on a video-conferencing setting like us now)
- Anne Glaser
    - If people are learning more than one coding language it’s easy to mixup syntax between them.
        - Explaining basic differences between them could help.
    - People advanced in the class are bored. People that are beginners are rushing through to keep up or not asking questions because they may feel intimidated.
        - If you identify the more experienced users, try to have them help the beginners near them if they’re comfortable.
        - Ask people questions to get them to talk more during the class and be more comfortable asking questions when it’s needed.
- Billy Broderick
    - A major difficulty is keeping the advanced folks engaged while not losing those with the least background. One possibility is to have those with more experience help those who are struggling (but you want to make sure they don’t miss anything).
- Dima Lituiev
    - Circulate surveys asking about learners’ expectations and requests
- Craig Boman
    - Could encourage more experienced attendees to help a less experienced person near them.
- Juliana Velez Lujan
- Julia Piaskowski
    - Keeping everyone engaged is far and away the most most difficult challenge, it can feel like no one is happy - people are bored, lost, confused or they have simply checked out. This can be the hardest part to manage.
- Nohemi Huanca-Nunez
- Saranya Canchi - Keeping the class engaged and ensuring continued interest in the lessons. Encourage the helpers to jump in and ask questions anticipating the learners may not always ask questions.
- Eva Sanchez. Would it be possible to have “the perfect pase” for students that are slow learners/beginners vs fast/more experienced students? Simple and clear explanations, team work.

Key Points

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

Feedback for morning of Day 2
https://docs.google.com/forms/d/11O4ZYI9Zoyf9QYNsxgJ73PbRrpxKUQNAHp4R1p96nXM

## Lunch

XV. The Carpentries: How We Operate

History

The Carpentry Community

Participating in the Carpentries: What's Your Role?
If you are at an in-person training, your instructor will hand out paper copies of a worksheet. If you are at an online training, you can get a digital copy at
https://carpentries.github.io/instructor-training/files/handouts/Carpentries_roles_worksheet_v2.pdf .
Working on your own, match up the roles with the descriptions. When you are done, think about the question at the bottom of the worksheet about what roles you might play, and enter your thoughts in the Google Doc.

https://carpentries.org/blog/2018/12/more-than-workshops/

Get Connected (3 min)

The Carpentry Community
There are many ways to get connected with the Carpentry community:

- Our websites are:
    - Software Carpentry https://software-carpentry.org
        - Blog https://software-carpentry.org/blog/
    - Data Carpentry http://www.datacarpentry.org
        - Blog http://www.datacarpentry.org/blog/
    - The Carpentries http://carpentries.org/
        - Blog http://carpentries.org/blog
        - Get involved (community overview) https://carpentries.org/community/
- Our lessons are hosted on GitHub; contributions to them and discussion of changes happens via GitHub pull requests and issues, and the lessons are published using GitHub Pages. More details are given below:
    - Data Carpentry on GitHub https://github.com/datacarpentry
    - Software Carpentry on GitHub https://github.com/swcarpentry
- Software and Data Carpentry share public discussion lists that host everything from lively discussion on teaching practices to job postings and general announcements:
    - https://carpentries.topicbox.com
- We publish a joint newsletter. https://carpentries.org/newsletter/
- Host monthly community calls and weekly instructor discussion sessions:
    - Check out our community calendar https://carpentries.org/community/#community-events

- You can also find us on
    - Twitter:
        - Software Carpentry on Twitter https://twitter.com/swcarpentry
        - Data Carpentry on Twitter https://twitter.com/datacarpentry
        - Carpentries on Twitter https://twitter.com/thecarpentries
    - Slack https://swc-slack-invite.herokuapp.com
    - Facebook https://www.facebook.com/carpentries/

Take a couple of minutes to sign up for the Carpentry discussion channels you want to stay involved with.

How a Workshop Works

- Materials
- Using the Names
- What's Core?
- Who Can Teach What
- Setting Up

Practice With SWC or DC Infrastructure (10 min)

Go to the Carpentries workshop template repository (https://github.com/carpentries/workshop-template) and follow the directions to create a workshop website using your local location and today's date. Put the link for your workshop website into the Google Doc.

Question and Answer (10 min)

What questions do you have about running and teaching at a workshop? Talk with a partner and enter your questions into the Google Doc.

A Culture of Contribution

Check Out the Discussion (5-10 min)

As an instructor, your voice is important! We want you to be actively involved in discussions about the lesson materials (and other aspects of the Carpentry community). Go to the GitHub page for the lesson you worked with over the past two days and click on the "Issues" tab. Read through some of the discussions and, if you have anything to add, please add it to the conversation! If you do make a significant contribution to the discussion, send a link to the issue to checkout@carpentries.org. Congratulations! You've just completed one of the three remaining steps in becoming a Carpentry instructor.

Key Points

- Carpentry materials are all openly licensed, but their names and logos are trademarked.
- Carpentry workshops must cover core concepts, have at least one certified instructor, use our pre- and post-workshop surveys and report attendance information.

XVI. The Carpentries: Teaching Practices

Organize Your Knowledge (5 + 5 min)

If you are at an in-person training, your instructor will hand out paper copies of a worksheet. If you are at an online training, you can get a digital copy here:
https://carpentries.github.io/instructor-training/files/handouts/Carpentries_teaching_practices.pdf

In groups of 3-4, fill out this worksheet, listing all of the teaching strategies and techniques you would use at each stage of a workshop. Be sure to include implementation details such as how long you would spend or how often you would do a particular strategy as well and any other information you think is important.
When you have a good amount of information in each box, type “done” in the Google Doc chat.

Code of Conduct Violations

Paste in GitHub Repository URL Here:

- EXAMPLE ~ John Simpson: https://symulation.github.io/2019-01-28-ualberta
- Katie Coburn: https://katiecoburn.github.io/test-test-test-umerced/
- Mara Sedlins: https://marasedlins.github.io/2018-12-12-csu/
- Jacob Deppen - https://deppen8.github.io/2019-01-01-seattle/
- Adriana Signorini: see UC Merced https://asignorini59.github.io/2018-12-12-Merced/
- Elizabeth Salmon : https://esalmon79.github.io/2019-01-01-Merced/
- Matthias Bussonnier for UC Merced :
    - https://meeseeksbox.github.io/2019-08-15-ucmerced/
- Jeffrey Weekley see UC Merced
- Sarvani Chadalapaka see UC Merced
- Chris Olson: https://cjolson64.github.io/2018-12-21-sio
- Martino Sorbaro https://martinosorb.github.io/2019-01-01-edin/
- André Palóczy https://apaloczy.github.io/2019-01-01-test/
- Anne Glaser: https://anneglaser.github.io/2018-12-12-ttt-practice/
- Billy Broderick: https://billbrod.github.io/2019-01-20-nyu/
- Dima Lituiev (removed the repo already;) i did not get it was only a demo in deleting
- Craig Boman https://craigboman.github.io/2019-05-30-miamiu/
- Juliana Velez Lujan
- Julia Piaskowski: https://jpiaskowski.github.io/pretend-swc-workshop/
- Nohemi Huanca-Nunez
- Saranya Canchi

XIV. The Carpentries: Workshop Introductions

Setting the Workshop Environment

Your Academic Past (5 min)
Think back to courses or workshops you really liked or didn’t like. How did those courses start on the first day? Were you confident in the instructors ability to teach the course? Did you feel like the instructor was enthusiastic about the course and invested in the students outcomes? Was it clear what you were going to be learning? Were you excited to

get the chance to be learning about those things? Or did you leave that first day thinking the instructor was uninterested, that you weren't the students they wanted to be teaching or you had no idea what the course was supposed to be about?

What's in a Introduction?

Get into small groups (3-4 people) and discuss these questions for 10 minutes.
Take notes on your answers – we'll combine them later.

What do you hope to accomplish in a workshop introduction?
What information do you need to include in an introduction to accomplish these goals?

GROUP NOTES:

UC MERCED
What do you hope to accomplish in a workshop introduction?
    Give people the agenda, gauge their levels, inform them of logistics
    Give them the overall workshop goal
    Share the code(s) of conduct
    Basically, use your checklist
    Explain why you're qualified for the workshop -- give them a motivation to stay
    Say the level of the workshop (what things it will cover); clearly set expectations
What information do you need to include in an introduction to accomplish these goals?
    Code of conduct
    Small elevator pitch
    Motivation and goals
    Icebreakers
Finally, compare your ideas with the list of topics below. Did you miss anything?
    Generally covered the list of topics provided
Did you come up with something that's not listed below?
    Not really

BREAKOUT ROOM 1

- What do you hope to accomplish in a workshop introduction?
    - Introduce other instructors
    - Discuss the goals of the workshop/learning objectives
    - Get to know the prior expertise of the students
    - Explain environment, set the tone
    - Set them up with their tools, give them prior instructions
- What information do you need to include in an introduction to accomplish these goals?
    - Results of the survey
    - Agenda -- list of contents and their schedule
    - Explain what carpentries is
    - Code of conduct (perhaps share email for issues)
    - Sticky note explanation for questions and task completion

BREAKOUT ROOM 2
Get into small groups (3-4 people) and discuss these questions for 10 minutes.

Take notes on your answers – we’ll combine them later.

What do you hope to accomplish in a workshop introduction?

- A good balance between giving people faith in your competence and presenting yourself as a learner along with them
- A good understanding of the day’s agenda and schedule
- Set a positive/welcoming/encouraging environment
- A sense that we’re all in this together / break the ice between participants
- Generate excitement for topic

What information do you need to include in an introduction to accomplish these goals?

- Intellectual background, code of conduct/establish a welcoming environment
- Summarize goals for the day
- Anecdotes on successes and failures
- Summarize schedule (so everyone knows when they have a break, can ea t and go to the bathroom)
- Set expectations: how to participate, when to ask questions, etc,
- Explain that you’re expecting people to follow along with live-coding
- Cover and explain code of conduct

UCSD

- What do you hope to accomplish in a workshop introduction?
    - Should be enthusiastic, establish your background
    - What they should expect, what is acceptable, what is not (including Code of Conduct)
    - Give an example of an end goal or learning objective
    -
- What information do you need to include in an introduction to accomplish these goals?
    - End goals and practical skills
    - Pre requisites and languages/skills being covered
    - Who to ask for help
    - Carpentries workshop format (sticky notes, helpers, up and down points etc)

### Putting it all together

---

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

Topics we’ve covered:

Here is a non-exhaustive list of topics:

- novice, competent practitioner, expert
- mental model
- formative assessment
- blind spot
- short-term and long-term memory
- cognitive load

- motivation
- demotivation
- error-framing
- life-long learning
- feedback
- lesson study
- Code of Conduct
- concept maps
- Multiple Choice Question (MCQ)
- Faded examples
- peer instruction
- going slowly
- “Just”
- Reach goals
- Avoiding dismissive language
- Reverse instructional design
- accessibility
- sticky notes
- one-up, one-down
- pre- and post-workshop surveys
- participatory live coding
- introductions

Help solidify these concepts and practices in your mind by linking and processing them now in aggregate. You can do this *either* as a **mind map on pen and paper**, or by adding them to **either of these two forms**:
https://carpentries.github.io/instructor-training/files/handouts/Carpentries_teaching_practices.pdf
https://carpentries.github.io/instructor-training/files/handouts/Wrap-Up-doc.pdf

Here’s an example of one that’s been filled out:
https://carpentries.github.io/instructor-training/files/handouts/Wrap-Up-doc-example.pdf

XVII. Afternoon Wrap-Up

Much of the material we have covered, in book form (assembled by Greg Wilson, freely available online):
http://teachtogether.tech/en/
http://third-bit.com/2017/05/31/how-to-teach-programming.html

Application form
https://amy.software-carpentry.org/forms/request_training/

The slug for this training is the name of this google doc
2018-12-11-ttt-online

Instructor Checkout
Instructor training checkout (checklist version): http://www.datacarpentry.org/checkout/
More detailed version: https://carpentries.github.io/instructor-training/checkout/

Schedule a Discussion or Demo (5 min)
Visit the discussion Google Doc (http://pad.software-carpentry.org/instructor-discussion) to sign up for a session. 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'd prefer to do your teaching demonstration before your discussion, visit the demo Google Doc (http://pad.software-carpentry.org/teaching-demos-recovered)and sign up there.

Final Feedback

One Up, One Down (5 min)
Provide one up, one down feedback on the entire two-day course.
Just as in our regular workshops, we collect post-instructor-training-workshop feedback. Your participation will help us evaluate the efficacy of this training and improve the content and delivery of the lesson materials.

- Katie Coburn - up
- Mara Sedlins - down
- Jacob Deppen
- Adriana Signorini -
- Elizabeth Salmon
- Matthias Bussonnier -
- Jeffrey Weekley. (had to leave) UP: The training was comprehensive and articulated just about everything you'd need to know or master. DOWN: The sessions were very long and the Zoom format was difficult at times to follow. The training would benefit from having a more integrated place to communicate, document and try hands-on.
- Sarvani Chadalapaka: UPs: Very useful session, learnt some very interesting concepts, comprehensive documentation so I can go back and refer whenever
- DOWN: Maybe do a shorter 3-day sessions instead of long 2-day sessions? Github part was a little fast-paced than other modules
- Chris Olson
- Martino Sorbaro
  - UP: excellent use of Zoom, much more seamless that I would ever have bet
  - DOWN: Maybe I missed it, but lack of general introduction on what the schedule of the two days was going to be, in particular explaining why the order of the modules was chosen.
- André Palóczy
- Anne Glaser: Up: Nice to be able to choose what we taught (R, git etc…) instead of very set examples. Down: Hard to find instructions during breakouts.
- Billy Broderick
- Dima Lituiev: (+): very comprehensive, good review of active learning and pedagogy with exercises. (-) I'd appreciate two shorter 5 or 10 min breaks instead of a long one each morning and afternoon -- it gets hard to keep concentration. (-) I would delegate concept learning to pre-home-work
- Craig Boman Up: appreciated the emphasis on research based instruction (Have to run! Thanks)
- Juliana Velez Lujan
- Julia Piaskowski: UP: loads of useful tips, training format both modelled and and facilitated a collaborative work environment. DOWN: we didn't have much time to prepare for workshop since email was sent late and

homework was difficult to complete in the time given - it was easy work, but I had family duties to deal with that nigh t

- Nohemi Huanca-Nunez
- Saranya Canchi - UP -> Lots of useful tips and philosophy of teaching. Very practical with the teaching demos and live coding sessions. DOWN -> Lots of material. Could have helped to have it sectioned into subtopics similar to the concept maps for the ideas to put it together. There were times when one group was still working on the exercises and the hosts were answering questions on the chat or moving on. There was not enough time at times for the exercises particularly one the first day, Could have helped to have some background prior to the workshop and focussed on only the core ideas during the workshop for a more focussed learning. There was a disconnect between the other participants and hosts since we were online and having some internet issues with the multiple platforms for communications.
- Eva Sanchez
  UP: Very motivating, great team work!
  DOWN: I was very tired at the end of the second day. One instructor was very dynamic and cheerful, the other instructor wasn't dynamic. Audio can be improved. A little bit more time for live coding may be needed. Could you change to a weekend instead of a weekday?

Ups (things that worked well or that you enjoyed)
Enjoyed live coding demonstrations (fun)
Few questions went unanswered (things were not passed over)
Passionate instructors
Lots of practical tips for teaching -- assessment techniques in particular (e.g. faded examples)
Training was comprehensive
Mental models was useful - good that things were documented
Zoom worked impressively well (20 people, breakout rooms worked well)
Good to switch topics or choose from things that are more familiar for me
Liked practical sessions for teaching
Liked feel of being in a site and building a community of teaching
Liked teamwork

Downs (things that could be improved for next time around)
Segments were long, so more quick breaks
A little hard to follow materials scattered over multiple places
Video setup could be improved (esp because of the split between groups and individuals, online and in person)
Sessions long and Zoom was occasionally difficult to follow - multiple spaces
Lack of connection with the online folks if in an on-site session (policy at UCSD if one remote, all remote)
More time for live teaching and introduction sections, hard to rotate among everyone within time.
Ran out of time on some sections (trim sections? Track with expected times)
Got email late, so there wasn't time to read it and prepare
Tough having homework overnight, esp without advance warnings
Got lost yesterday (no prior warning) - there was a lot of material especially in early sessions -- idea: focus in on one core idea and deep dive on those
Time zone differences make it harder (idea: weekend workshops)

Minute Cards (5 min)
In addition to giving one up, one down feedback. Please also fill out your sticky notes to give your instructors anonymous feedback.

https://docs.google.com/forms/d/1Vj-vJCylc8kQSkyQBy9VZ6g96H06fm5UaB4nzjn2Hzs/edit?ts=5c116a15

Post Workshop Surveys (5 min)
Assessment is very important to us! Please take 5 minutes to complete this five-minute post-workshop survey. (https://www.surveymonkey.com/r/post-instructor-training)

Key Points

- 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.