Welcome to The Carpentries Etherpad!

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try https://etherpad.wikimedia.org).

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

----

Course website: https://murraycadzow.github.io/2020-10-14-ttt-online-nz/

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

- Kiera Tauro (She/her), University of Canterbury, kiera.tauro@canterbury.ac.nz, @librarykiera
- Brian McElwaine (he/him), University of Canterbury, brian.mcelwaine@canterbury.ac.nz
- Megan Guidry (She/ her), New Zealand eScience Infrastructure, megan.guidry@nesi.org.nz
- Kim Pham (she), University of Denver, kim.pham60@du.edu
- Paul Melloy; (he/him)The University of Southern Queesland, paul.melloy@usq.edu.au @PaulMelloy
- Leticia Castro (She/her), Victoria University of Wellington, leticia.castro@vuw.ac.nz
- Kari Clifford (she/her) (Otago Medical School Dunedin Campus (Surgical Sciences) Kari.clifford@otago.ac.nz)
- Lara Urban (she/her), University of Otago, lara.urban@otago.ac.nz
- Olivia Angelin-Bonnet (she/her), Massey University, angelin-bonnet@massey.ac.nz
- Mark Crowe, QCIF, Brisbane (he/him) mark.crowe@qcif.edu.au
- Swabir Silayi(he/him), George Mason University, ssilayi@gmu.edu
- Annabel Whibley (she/her), University of Auckland, annabel.whibley@auckland.ac.nz, @alt_allele
- Upendra Raj Bhattarai (he/him) University of Otago, Dunedin, bhaup057@student.otago.ac.nz
- Patricia Haden she/her University of Otago Scientific Programming
- Aurelie Laugraud (she/her) AgResearch

- Aleksandra Pawlik she/her, Manaaki Whenua Landcare Research pawlika@landcareresearch.co.nz
- Arindam Basu, he/him, University of Canterbury, arindam.basu@canterbury.ac.nz
- Paula Andrea Martinez, she/her, @orchid00

Please fill out the pre-training survey at https://www.surveymonkey.com/r/instructor_training_pre_survey?workshop_id=instructor-training

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

I. Welcome

Code of Conduct:
https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

Introductions

Information for Today’s Learners

1. Add your name to the Etherpad above
2. 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 Etherpad, 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.

A Brief Overview of the Carpentries
https://carpentries.org/workshops/

Instructor Training Workshop Overview

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

Assessing Trainee Motivation and Prior Knowledge

Background (3 min)
Have you ever participated in a Software, Data or Library Carpentry Workshop?

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

Which of these most accurately describes your teaching experience?

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

similar activities.

Key Points:

- The Carpentries 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 the Carpentries.
- Learner motivation and prior knowledge vary widely, but can be assessed with a quick multiple choice question.

II. Building Skill with Practice
https://carpentries.github.io/instructor-training/02-practice-learning/index.html

The Carpentries Pedagogical Model

Acquisition of Skill
https://carpentries.github.io/instructor-training/fig/skill-level.svg

- Novice
- Competent practitioner
- Expert

Cognitive Development and Mental Models (5 min)
In the Etherpad, 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?

- Olivia: domain: modelling of regulatory networks; concepts: molecular biology (main steps of gene expression, different types of regulatory interactions), mathematical modelling (e.g. deterministic vs stochastic), equations for biochemical reactions; links = how to translate our knowledge about a biological system into equations, detail of the modelling vs detail of the biological mechanisms
- Lara: domain: statistical genomics: concepts: association studies (linear models with confounding factors), hertiability of traits/prinicples of inheritance/phenotypic and environmental variance, genomic predictions (Bayesian modelling, phenotypic predictions based on genomics); facts: DNA, DNA extraction, PCR, sequencing, data processing, uncertainty/error estimates
- Upendra: domain:Genomics and transcriptomics: understanding of the software pipelines and algorithms, gene regulations, functional genomics, genome and transcriptome sequencing, analysis and their intrepretations
- Brian: Domain- information management. Key concepts include things like information providence, governance, sourcing, managing information, synthesising new meaning. Relationships exist between entities and information sources. It also relates to information flow within and between organisations.
- Paul: Domain - Agricultural science; Key aspects to plant pathology that help fram and understand the work is the 'disease triangle' which links 3 aspects together, presence of a virulant pathogen, susceptible host and a conducive climate. If you have these three aspects you get disease on the plant. I use this to give background to link how modelling can help us understand the disease, we

use aspects to predict the risk of disease.

- Kari: Domain: Surgical Sciences, we have a number of factors that can affect patient outcomes following surgery, so we can use mental models to look at the associations between these, and which ones have the most impact.
- Patricia: Software development. Architectural principles such as abstraction, encapsulation, coupling, etc.
- Swabir:Domain- Materials Modeling and Simulation; things to think about include interaction forces and parameters and the methods of simulation (e.g. time integration/monte carlo simulations); Connections that need to be made would be translating the mathematical expressions(integrations) into code and how to get the correct parameters for the interactions.
- Kiera: Domain: information literarcy/library stuff. Concepts: Finding and referencing appropriate academic material, how to store and manage information, some research data management, finding where to publish, research metrics.
- Aurelie : bioinformatics : translating in computer language what we want to find from biological data.

Kim: domain - library IT/software management. concepts- software development, systems architecture, digital preservation. use models of information systems design/architecture, web standards and specifications, project management strategies

Annabel: evolutionary genomics: Concepts: principles of genetics/genomics/evolutionary theory; bioinformatics and HPC environments, reproducible research and data-sharing. Facts: sequencing technologies, assembly algorithms, gene prediction, functional annotation, comparative frameworks, population analysis

List a type of formative you are aware of:

- Olivia: small worked example during a lecture, tutorials, quiz at the end of a chapter
- Kiera: essays, forum posts, projects, quizzes
- workshops with exercises, quizzes
- Upendra: Quiz,
- Laboratories/ward rounds with clinicians
- Paul - at Uni workshop/tutorial questions and challenges which are answered in class with the help of peers and the tutor/instructor
- Brian: Developing code in pairs, and reviewing work. Quizzes during term addressing what has just been learned. Short essay questions on a relevant topic that aren't graded.
- Patricia: Marked exams; final presentations without feedback loop are summative. Projects marked with feedback are formative, as are practice presentations with feedback and opportunity to change
- Leticia: short practice questions, reasonably frequently
- Lara: short summary of what was taught
- pop up quiz?
- observations

Formative Assessments Come in Many Forms

Identify the Misconceptions (10 min)

Choose one of the wrong answers to the question below and write in the Etherpad what the misconception is associated with that wrong answer.

Q: what is 27 + 15 ?

a) 42

b) 32

c) 312

d) 33

Formative Assessments Should be Frequent

How Many? (5 min)

The Carpentries use formative assessments often. How many have we done since the start of this workshop? Put your guess in the Etherpad along with one example and the purpose that assessment served.

- Olivia: 5 - examples of formative assessment - to check if we understand the difference with summative assessment
- Kari: 7, this one: how many formative assessments and give one with a purpose. The purpose would be to keep us awake and make us think about the content & hopefully remember this stuff..
- Paul - 6...
- Leticia: 3ish
- Brian 3 or 4, on the topic?
- Lara: 3
- Swabir: 5
- 6? multichoice discussion
- Kiera: 10?
- Upendra :5?
- lots
- Kim: 5

Session 2: Expertise and instruction

Who is an expert? Think of an expert in your field

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. Expertise and Instruction

https://carpentries.github.io/instructor-training/03-expertise/index.html

What Makes an Expert?

What Is an Expert? (5 min)

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? 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?

- Olivia: J. Marshall (lecturer), expert in R programming, will always have an idea on how to solve any problem when it comes to programming
- Kari:OUr supervising clinician; I value his expertise but also his patience in explaining concepts to me and helping me work through problems. Not sure I am an expert in anything.... but when I feel expert I am more confident and efficient in performing tasks.
- Paul - Levente Kiss in Powdery Mildew taxonomy; vast knowledge and experience on the group
- Leticia: My boss! Can ask her complex questions about sometimes not very well thought through concepts, and she can extrapolate and answer by linking main points in the question. Can contextualise many random concepts.
- Brian: I used to work with our "Library IT" department. 4 people, 1 novice, 2 competent, 1 expert. The Expert knew the context of every single problem that came up. He had such vast experience under his belt that he was like a walking StackExchange for our particular set of systems. He was able to make connections about how apparently disparate systems affected each other- things you wouldn't ever know unless you'd experienced all the problems. All the problems.
- Lara: Alana Alexander - she has a response to anything genomics-related, it might not be that she could actually answer all questions without googling, but she has a very good overview of the field and of possible approaches; I am not an expert in a whole field yet since I just switched fields, but I would say that I am an expert in quantiative genomics approaches - since I could advise others on how to run their analyses.
- Swabir:D.A Papa, expert in density functional theory - able to computationally solve most problems in the field.

- Kiera: Anton for data/carpentries stuff - if he doesn't know the answer (which he often does) he'll find out
- Upendra :Alana Alexander and Hughes. They have lots of experience on the field and they actually know how to teach and are very approchable
- Annabel: Pretty much everyone I've encountered via NeSI technical support. Specialist subject knowledge, good professional network for directing queries onwards, understanding the essence of (sometimes unclear) question.
- Kim: Diego - in our Islandora open source community knew the different components and how they worked
- Patricia: Grady Booch - can see an entire problem architecture at once . Less expert practitioners head down wrong paths and have to backtrack when they get stuck.
- Aurelie : Pf Gilles thomas: lots of knowledge and ability to use this knowledge to understand and make conections

Connections and Mental Models

Limitations of Expertise

Fluid Representations (5 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.

Kiera: I've used an example for getting people to understand the difference between a journal/volume/issue/article as a tv show/season/episode/time stamp
Brian: Information sources like authors and organisations can be sources of honest communication, but also sources of bias that the reader needs to consider, and positions within a field can have a big impact on how sources are represented.
Kim: describing our digital collections ecosystem - to our users describe different components as backend/frontend/descriptive names for our systems simplified based on how they use it. different from the way we describe it in our own IT team, described more as the software that performs what functionality at what stage/how they are integrated
Paul: Plant disease modelling could sometimes be described as applied research to use existing knowledge about a host-pathosystem and apply mathematics to describe the relationship and predict the risk of eppidemics; sometimes I would provide an example of the application of my work
Leticia:when teaching cell culture technique and maintenace of cell lines, explaining what the maintenance entails and likening it our own maintenance ie. feeding, rest etc.
Kari:We provide exercise training to patients before surgery. I explain to older people how they can increase their fitness. Some approaches work better in some than others...So for some they need to work until they are "puffing", and others like to look at their heart rate.
Lara: QQ-plots vs Manhattan plots to represent P-values in genomic association studies; switching between GUI and command line while programming
Olivia: gene regulatory networks: I can think of it in the biological sense, i.e. biochemical reactions, or the mathematical representation, i.e. rates of transcription etc, or in the data sense, i.e. what do we measure to get to them - in a Carpentry lesson: data-frames: explaining that you can view it as columns vs rows or variables vs observations
Swabir:optimizing/scaling code performance on supercomputers
Annabel: genome comparisons- different levels and conceptual frameworks to do this.
Upendra: genome assembly: Understanding of different sequencing data and ways to incorporate them vs understanding of the software algorithms
Patricia: alpha level vs p-level vs confidence level in hypothesis testing
Aurelie: genome assembly seen as book shredding exercice and putting sentences back together

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 explaination below.

Kiera: This thing doesn't have enough information available for me to reference it! Explaining what the purpose of a reference is, how you can find more information about a thing (or use the information that's already there) and adapt it accordingly
Brian: "Ive searched for my essay topic but there's nothing in the library"
Kim: object not found. Looks like the asset doesn't exist in the repository. An issue might be that you don't have the correct link address, or that the repository doesn't think that it doesn't exist anymore, perhaps it's not indexed in the search engine
Paul: (R) missing argument when none supplied; missing ')'; unknown function `rast`
Leticia:leaving a certain flag off or wrong flag to a basic command that results in a random answer, not at all what i intended
Kari:from my R studio: Error: unexpected ',' in "   color = "variable","Every. Single. Day. I would

explain to a learner to a) take time when typing, and b) don't get too frustrated, is a common mistake!
Lara: switching between R and Python and using the respectively wrong functions for the loop, printing, etc; I guess my recommendation would be to learn one of these two languages properly and to use it most of the time ;)
Olivia: in R, error when using the select() function, the error message is not informative but I know it's because Ioaded another package with a different select function after I loaded tidyverse
Swabir:not using the correct enviroment when running software programs - what does this error mean.
Annabel: command not found. often to do with what programs are present in the software environment, would explore how to check what is there and how to correct if program is missing.
Upendra: typos in command line, syntax error. be careful when you type,
Patricia: object not found (we remind students that you have to spell it exactly the same way *every time*)
Aurelie:typos

Expert Awareness Gap (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?

Kiera: This is a personal life example, but I've been mountain biking for a couple years now, and have been consistently slow on a certain type of track, and it wasn't until the person who was teaching me asked a question about what I was doing that they realised that something they had always done and was obvious to them was the complete opposite of what I was doing. Doing it the new way was a 100% improvement
Brian: I'm learning about bibliometrics to support our researchers in this area, but it's really hard to get someone experienced in this area to give me the idiot-proofed Brian-friendly version, and they seem to have trouble explaining some of the deeper concepts of why we do what we do and what we're really demonstrating.
Kim: learning a new system we inherited when the maintainer left. in order to work with the API i have to understand more about the data model, domain + terminologies (system stores archival finding aids)
Paul: When teaching genetics there was a excercise on linked trait. The example we used was why would flower colour and seed texture traits be linked. After providing the answer I still had to explain to a student that flowers (in most cases) produce seeds. I expected all students understood this simple fact that flowers produce seeds
Leticia:trying to answer questions by reading documenation...very rarely helpful. Jargon is my enemy. Not from a compuational background so using words that have no meaning to me don't help. I prefer to source answers from actual people's experiences
Kari:My basic degree is in nutrition. Sometimes I have to work hard to remember that the average person doesn't have the same knowledge as me , and it actually puts me off having casual conversations about the topic sometimes (always happy to teach it in a formal environment).
Lara: Identifying a list of applications that can be used for a specific task - I guess it requires efficient literature research which also has to be learned
Olivia: I am going to be a lecturer next year, so I'm trying to understand all the things to do to prepare a course, etc, and when talking to other lecturers sometimes they jump between subjects and I don't understand the connection
Swabir:making the constant mental shift switching between teaching students with almost no concept of computing and working with expert researchers
Annabel: Trying to help my daughter learn piano- it's so easy to forget all the conscious steps that I made when learning and that I need to break down again in order to help.

Upendra:I am trying to figureout the proper sequencing technique for my transcriptome analysis, there are so many basic things I need to understand
Patricia:My violin teacher doesn't have to look at where she puts her fingers or how she holds her bow
Aurelie:some poeple I plya music with don't see how I still need the music to play,

The Importance of Practice (Again)

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.

IV. Memory and Cognitive Load
https://carpentries.github.io/instructor-training/05-memory/index.html

EXERCISE Test Your Working Memory (5 min)
This website https://miku.github.io/activememory/ implements a short test of working memory. In this test, you will see about twenty words, each for a short amount of time. Try to memorize as many as you can.

What was your score? Write your answer below.

- 9
- 7
- 8 (2 the first time ;p)
- 10 :'(
- 4 and then 8 because I was sad I did a bad job the first time
- 6
- 5-6
- 6
- 11
- 8
- 5
- 6

Concept Maps as Instructional Planning Tools

Example concept maps:
https://carpentries.github.io/instructor-training/fig/array-math.png
https://carpentries.github.io/instructor-training/fig/conditionals.png
https://carpentries.github.io/instructor-training/fig/create-destroy.png
https://carpentries.github.io/instructor-training/fig/dict-set.png
https://carpentries.github.io/instructor-training/fig/io.png
https://carpentries.github.io/instructor-training/fig/git_concept_map.png

https://carpentries.github.io/instructor-training/fig/lists-loops.png

Concept Mapping (10 min)

Create a hand-drawn concept map for a part of a Carpentries 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*” in the Etherpad chat once you have finished.

Why Guided-Practice is Important

cognitive load

- ***Intrinsic*** load is what they have to keep in mind in order to carry out a learning task.
- ***Germane*** load is the (desirable) mental effort required to create linkages between new information and old (which is one of the things that distinguishes learning from memorization).
- ***Extraneous*** load is everything else that distracts or gets in the way.

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):
    ____
```

EXERCISE 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 Carpentries lesson, or from another lesson you have taught recently.
2. Replace 2-3 pieces of the code with a blank.
3. Write a question to test the student’s ability to correctly fill in that blank.
4. Paste your faded example in the Etherpad.

Upendra, Brian, Lara:

```
%matplotlib inline
import matplotlib.pyplot as plt
import pandas as pd

data = pd.read_csv('data/gapminder_gdp_oceania.csv', index_col='country')

# draw a line
years = data.columns.str.strip('gdpPercap_')
data.columns = years.astype(int)
data.loc['Australia'].plot()

# draw a barplot
plt.style.use('ggplot')
data.T.plot(kind=___)
plt.ylabel('GDP per capita')

# add Australian data to line
years = data.columns
gdp_australia = data.loc[___]
plt.plot(years, ___, 'g--')

# draw a scatterplot of Australia vs NZ
plt.scatter(___, ___)

# plot a line and scatterplot of Asia
data_asia = pd.read_csv('data/gapminder_gdp_asia.csv', index_col='country')
```

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 Skill with Feedback
https://carpentries.github.io/instructor-training/06-feedback/index.html

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

Minute Cards

One-Up, One-Down

Give Us Feedback (5 minutes)
Write one thing you learned this morning that you found useful on one of your sticky notes, and one question you have about the material on the other. Do *not* put your name on the notes: this is meant to be anonymous feedback. Add your notes to the pile by the door as you leave for lunch.

Key Points

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

https://docs.google.com/forms/d/e/1FAIpQLSdrr6rCcQrghivW5frl_k2j46ce1M8cdbVnHJgnGh0rn7ax-g/viewform

===DAY 2========================

VI. Motivation and Demotivation
https://carpentries.github.io/instructor-training/08-motivation/index.html

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
https://carpentries.github.io/instructor-training/fig/what-to-teach.png

Authentic Tasks: Think, Pair, Share (15 min) (till about 13:05 NZST)
Think about some task you did this week that uses one or more of the skills we teach,
(e.g. wrote a function, bulk downloaded data, built a plot in R, forked a repo) and explain how you would use it (or a simplified version of it) as an exercise or example in class.
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 Etherpad, Share the task and where it fits on the graph.
As a group, we will discuss how these relate back to our "teach most immediately useful first" approach.

Kiera: My task (getting data from SciVal) is low usefulness but also a low amount of time to master

Kari: generating simple boxplot- useful and quick to learn. I would start by showing an example of one to get them thinking about it/inspired
Brian: Also getting data from SciVal (a service that aggregates information about academic performance, used for example in QS rankings. I think this data itself could be used to make a meaninful dataset for a carpentry on manipulating data, targetted to the department within UC that I was working with. I think working with real data affecting their careers would be motivating and help people to understand the value of thinking about data innovatively. I think it could also lead to conversations about what to ignore- deciding for yourself what's in the too-hard basket.
Olivia: the task is to create a plot from a dataset in R. As an exercice, I would only show the main functions to get the plot, i.e. ggplot(), geom_point or geom_boxplot, etc., which would sit on the top left corner of the graph. I would leave out from the exercice all the details that make the plot look nice, i.e. changing the colours, the background, centering the title, etc (at the bottom right corner of the graph)
Upendra: my task (Using multiple softwares as a pipeline to scaffold the genome) I think trying single software at a time and evaluatinng the output individually will give more information about what what you are doing and how can you improve for better. Than investing more time to figure out the dependencies and creating the pipeline.
Leticia: plotting with python on a jupyter notebook (mastering python and just getting around on the notebook). For beginners, probably in the middle for time to master but very high up on the usefulness scale.
Lara: 1. plotting in R/python: very important (good for understanding data and for motivation), easy/medium to learn (depending on experience with programming language); 2. command line usage: very important and easy to learn (just a few commands necessary at the beginning); 3. job submission on HPC clusters: very important if necessary, but difficult to learn, especially memory usage/parallelisation
Swabir: moving from statements in python to creating functions so that the same process can be exwcuted without redoing everything. This is a useful skill and not difficult to master as well.
Kim: My task is writing a script to download data from Instagram. This is low usefulness and medium time to master. Low if you are trying to learn the basics of Python, it requires usage of libraries and you are not doing as much basic Python programming. Medium time to master - understand some Python but also have to understand how to use libraries and how the library works.
Paul & Patricia: Commenting code requires almost no time to master and has a huge payoff for usefulness

Why Do You Teach? (till about 13:23 NZST)
We all have a different motivation for teaching, and that is a really good thing! The Carpentries wants instructors with diverse backgrounds because you each bring something unique to our community.
What motivates you to teach? Write a short explanation of what motivates you to teach. Save this as part of your teaching philosophy for future reference.

Olivia: I want to share my enthusiasm for R and show people that you don't have to be an expert in programming to be able to use it for your research (specially wet lab scientists who may not have received any training but would like to analyse and visualise their data). Also, I want to be a good lecturer and make sure that my students receive a good formation.
Kari: I love connecting with students, and I am motivated to teach to share my skills with others and develop my own capacity to a) understand the subject matter b) communicate effectively.
Brian: I enjoy teaching, and I want to be better at it. I believe that university libraries are likely to be the people that have to deliver the carpentry-type skills, so I want to be involved early with it if I can.
Leticia: I enjoy sharing my knowledge and helping people learn in a comfortable environment (learning from someone who has gone through a similar journey). Having struggled so much when starting out due to minimal resources, i want to help others so that they don't have to battle so much

Upendra: I used to work more in wet lab before and when I was trying to learn bioinformatics I took few data and software carpentaries and those classes helped me a lot to understand basics and advance my knowledge further. So as now I have bit more experience I would like to share my learnings and experience with others.
Kiera: I like helping people, and most of the things I teach are practical, and can immediately impact the way students go about their assignments etc.
Paul & Patricia: Computation skills are increasingly becoming a required skill for my research discipine and should be taught to improve the quality of science through improved reproducability.
Swabir: Working with researchers that are not traditionally expected to be familiar with programming concepts, so the Carpentries would be a good way to introduce them to the concepts.
Kim: I am motivated by building community, and teaching brings people together who are interested in the same topic/learning the same skill. It also allows for future collaboration opportunities with more people. I think teaching also helps solidify concepts for me, I always find that I learn new things by teaching if I see it from a new perspective with other people.

Things You Shouldn’t Do in a Workshop (till about 13:57 NZST)
(think about your demotivational experience as a student or teacher - look at the list below and tick off any of the bulletpoints you recognize as familiar in your story)

- Tell learners they are rubbish because they use Excel and/or Word, don’t modularize their code, etc.+
- Say negative things about Windows and praise Linux, e.g., say that the former is for amateurs.+++++
- Criticize GUI applications (and by implication their users) and describe command-line tools as the One True Way.+++
- Talk contemptuously or with scorn about any tool. Regardless of its shortcomings, many of your learners may be using that tool. Convincing someone to change their practices is much harder when they think you disdain them.+++
- Dive into complex or detailed technical discussion with the one or two people in the audience who clearly don’t actually need to be there.++++
- Pretend to know more than you do. People will actually trust you more if you are frank about the limitations of your knowledge, and will be more likely to ask questions and seek help.++
- Use the J word (“just”) or other demotivating words we talked about in a previous lesson. These signal to the learner that the instructor thinks their problem is trivial and by extension that they therefore must be stupid for not being able to figure it out.++++++
- Take over the learner’s keyboard. It is rarely a good idea to type anything for your learners. Doing so can be demotivating for the learner (as it implies you don’t think they can do it themselves or that you don’t want to wait for them). It also wastes a valuable opportunity for your learner to develop muscle memory and other skills that are essential for independent work.++++++++
- Feign surprise. Saying things like “I can’t believe you don’t know X” or “You’ve never heard of Y?” signals to the learner that they do not have some required pre-knowledge of the material you are teaching, that they don’t belong at the workshop, and it may prevent them from asking questions in the future. ().++ <-- makes me think of this https://xkcd.com/1053/

Look at the accessibility posters (till about 14:18 NZST)
https://github.com/UKHomeOffice/posters/blob/master/accessibility/dos-donts/posters_en-UK/

accessibility-posters-set.pdf
1. list 3 things that you learnt (as a group - so 3 things per group)
2. pick a representative who will report back from the group
PatriciaZyangKim: 1. designing for a range of accessibility needs 2. some needs are contradictory e.g. contrasting colours 3. dyslexia design - use fonts

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

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 (p), effort-based (e), or improvement-based (i) praise?

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

VIII. Teaching is a Skill
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 Carpentries-style workshop teaching and providing some feedback.
Watch this example teaching video (https://www.youtube.com/watch?v=-ApVt04rB4U) as a group and then give feedback on it. Put your feedback in the Etherpad. Organize your feedback along two axes: positive vs. negative and content (what was said) vs. presentation (how it was said).
Content +shows his errors and how he fixes them
Content - unsure whether his audience knows anything about what he's doing, doesn't explain any of itskips over some content with little explanation
Presentation + makes some eyecontact
Presentation -dissmissive, CHECKS HIS PHONE :(, too fast rude at the start
Paul: (-)Presentation turns away from the audience to explain something, (+) takes through the explaination in a step by step manner; content justifies a action by saying "right" expecting the audience to understand what he has done
Kiera: Presentation (Goes very quickly, condeceding and dismissive, makes an error, but doesn't explain what that error is, and just goes on to fix it quickly without explaining what he's doing) Content (Lots of small writing on a big screen, assumes people have knowledge without really explaining things)
Leticia (content -ve): makes assumptions about people's knowledge
presentation -ve: repeatedly refers to 'simple' content not everyone may be familiar with
Olivia: Content + : gives examples. Content - : does not explain everything, errors are not explained.
Presentation + : summary of what has been presented in the last session. Presentation - : talking fast,

small font on the screen, so it's hard to read; "very easy", "of course" and other dismissive language

Patricia:
Needs to avoid hostile tone at start.(-pres)
Vocabulary used probably not appropriate (based on level of content) (-content)
"Trust me"" indicates that lesson is not properly constructed -- if there is something they need to know, explain it (- content)
Turn off your phone when teaching unless someone is actively monitoring an emergency or something; not to do so is disrespectful (-pres)
Failing to explain the "fix"misses a useful opportunity for content rehearsal (-content)

Upendra: Presentation+: shows errors

- Presesntaion-: too fast, fonts very small, checks on mobile phone
- Content- : couldn't understand

Feedback on Yourself (25 min) 15:47-> 1552 prep, then present. all wrapped up by 16:15 (NZDT)

1. Split into groups of three.
2. Individually, spend 5 minutes preparing to teach a 90-second segment of the lesson episode you chose before the start of the training course.
3. Take it in turns to teach for 90 seconds and then everyone gives feedback (including the person who taught) then rotate person. Use a whiteboard or other visual aids if available (Note: Do not live code you lesson. There will be a chance to live code later.) Keep a strict time limit of 90 seconds per person (one person should be responsible for the timekeeping).
4. After everyone has given feedback on all of the presentations, return to the main group and put everyone’s feedback about you into the Etherpad.

- 90 seconds worth of content
- A minute and a half

Patricia (dataframes): very well explained; try to avoid any technical terms
Leticia (wildcards): very good usage of repetition (slightly differing examples illustrating underlying functionality) ; define "jargon" terms
Lara: Super motivation of task; genomics is hard
Brian: (dates as data): decent pacing, content is awkward, good attempts at engagement.
Kim did a great job explaining git and git hub. In particular she spoke slowly and clearly, and defined the concepts before moving on.
Patricia: awesome ability to put context around a concept. Used some techincal terms that may lose novices.
Lara: great ability to put goal into perspective ie. why do we want to do what we're about to do.
Annabel: (introduction to working in terminal). Pacing and content clear, good explanation of commands (including spelling each out), question about whether technical terms would be familiar to audience, could have justified motivation more
Kari did a great job setting up the context for what we were going to learn, went at a good pace to follow along, did not use jargon or acronyms when describing exercises

Kiera: Good enthusiam, and not too much detail. Good leading with what's relevant, but might have used too much jargon - make sure to explain

----

If you would like a trainer to pop in to clarify anything pop a message here with your room number
Room 4: You mention videos in item 2 above. You want us to record these? not recording
k thanks
Room 5 : pace is important, let people have time to absorb what we say. Focus on one message to make it stronger . Important message first. People gave very clear explanations and show great knowledge of the topic they chose.

Day 2
https://forms.gle/csxfdThYBQ2SE2n4A

What questions do you have about running and teaching at a workshop? List them below and we'll address them through day 3 and 4

X. Welcome Back
https://carpentries.github.io/instructor-training/13-second-welcome/index.html

Day 3 sign in
Patricia Haden - Scientific Programming University of Otago
Brian
Olivia Angelin-Bonnet, Massey University
Aurelie
Kiera Tauro
kari Clifford
Swabir Silayi
Kim Pham
Leticia Castro, Victoria University of Wellington, Biological Sciences
Paul M
Upendra Raj Bhattarai, University of Otago
Annabel Whibley, University of Auckland
Zoe Yang ESR
Lara Urban

Paula Martinez - helper

Questions (5-10 min)
Last session we asked you to read some resources about the logistics of teaching and running Carpentries workshops. Please add your questions about logistics and preparation to the Etherpad. We will answer these questions in the etherpad during your work time and will return to this list later today.

Olivia: is someone from Carpentries monitoring the fees that participants are charged?

Paul: How do you make the minute cards for in person, and online feedback?

- -Both are anonymous, for in person we use sticky notes, for online feedback is usually a google form (or any other form tool you use) with 1 question "what can we improve?", question 2 "What did you like/enjoy the most?"

Leticia: do you have to wait to hear back from carpentries after registering a self directed workshop, before going ahead?

- - Recommended, yes. You can contact them again if you don't hear in 2-3 days

How much can you add/subtract to lessons?The criteria have 3 lessons listed per program

- - Now we have what it is called Mix and Match https://carpentries.org/blog/2019/09/Workshop-request-forms-and-more/ you can select what lessons to teach.
- "If you are teaching a portion of The Carpentries curriculum or if the workshop does not align with the Core Curriculum we ask that you still register your workshop and select the “Mix & Match” option for the question “Which Carpentries workshop are you teaching?”. *This option is only available for Self-Organised workshops. Centrally-organised workshops are required to follow the Core Curricula.*"
- https://carpentries.org/workshop_faq/

Annabel: Is it easy to set up an etherpad? Any particular sites with good functionality (or ones to avoid)?

- - https://pad.carpentries.org is for carpentries workshops.
- - There is also https://etherpad.wikimedia.org/ for general purpose.
- - You can use other tools like hackmd, codi.md if your audience is confortable with markdown, or google docs, only thing to keep an eye on is if you have from than 20 ppl, it slows down quiet a bit. So you might want to break it in parts.

I made a contribution in the past before instructor training does that count towards my instructor checkout, same question about teaching a full Carpentries lesson

- - Probably best to double check with checkout@carpentries.org

Check out process
https://carpentries.github.io/instructor-training/20-checkout/index.html
If you haven't yet please fill in the form https://amy.carpentries.org/forms/request_training/

Patricia: For suggested edits to the lesson documents do you prefer one PR per edit, or multiple edits in a single PR?

- - if it is a single file (.md) you can send a PR usually for smaller changes like typos
- - if it is a big change and over many files best to write an issue first.

In small groups, read and discuss one of the three checkout procedures listed above and described in detail at https://carpentries.github.io/instructor-training/checkout/. Make notes in the Etherpad and when you’re done, report back to the full group about the requirements for that stage of the process. What questions do you still have about the checkout process?

This exercise should take about 5 minutes. back at 12:55 (NZ

Break out room 1
What's the difference between pre-approved applications and open applications? Which are we likely to be?

Break out room 2
We are a little unclear about the mechanics of the group discussion and demo lesson. Who arranges that? Is the demo lesson a simulated one or do we need to arrange real learners? How do we find a discussion, etc.? Thanks.

  - - look below for pad URLS
  - - Demo: Prepare to teach a full Carpentries lesson (i.e. the content of one lesson repository). Then perform a 5-minute live coding demo for that lesson starting at a point chosen by the session lead.

Break out room 3
Once we're certified, how often do we have to teach to keep the certification.

- - teach at least once a year to maintain voting status
- - Currently it's "once badged always badged"

Do we have a redo the discussion if the discussion leader doesn't think we've participated adequately and what s adequately (sounds pretty subjective)

  - - Participating is being in a discussion and asking and answeing questions.

- Pad of pads ★
https://pad.carpentries.org/pad-of-pads
https://pad.carpentries.org/community-discussions -- schedule this one first,.
(join other discussions to keep updated on changes, ask questions, provide feedback, etc)
https://pad.carpentries.org/teaching-demos -- schedule this one after the discussion
for times check the URL See the start time in your time zone

Carpentries Handbook★
https://docs.carpentries.org

Running workshops
https://docs.carpentries.org/topic_folders/hosts_instructors/index.html

Get Connected (3 min)
Take a couple of minutes to sign up for the Carpentries discussion channels you want to stay involved with.

There are many ways to get connected with the Carpentries 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/
- Library Carpentry https://librarycarpentry.org
- Blog https://librarycarpentry.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
- Library Carpentry on GitHub https://github.com/LibraryCarpentry
- The Carpentries 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
- Library Carpentry on Twitter https://twitter.com/LibCarpentry
- Carpentries on Twitter https://twitter.com/thecarpentries
- Slack https://swc-slack-invite.herokuapp.com
- Gitter https://gitter.im/LibraryCarpentry/Lobby (Library Carpentry)
- Facebook https://www.facebook.com/carpentries/

Carpentry connect interested:
Brian (brian.mcelwaine@canterbury.ac.nz)
Kari (kari.clifford@otago.ac.nz)
Upendra (bhaup057@student.otago.ac.nz)
Leticia (marialeticia.castro22@gmail.com)

Live coding (back at 14:46)
List some advantages and challenges of participatory live coding from both a learner's and an instructor's point of view in the Etherpad. You can add +1 if you agree with an item already listed

Benefitial about live coding

- You can see mistakes etc. made by the instructor so it looks more achievable+1
- Better way to miprint the lesson on yourself
- Corrections in real time
- Getting used to errors
- DIrectly involved in problem-solving
- Better understanding of logical structure of the code
- Understanding importance of each parameter by typing yourself

Negative about live coding

- Might be easy to get lost if the pacing isn't right
- Cognitive load issue for instructor

- Boring for people who can already do the content, but they have to stay
- There's no safety net in things go wrong
- Slowing down if some tools/programs don't work
- Other distractions
- Not as efficient

Instructions
------------------ back at 15:08

Watch this first participatory live coding demo video (Video 1) and this second demo video (Video 2) as a group and then summarize your feedback on both in the Etherpad.
Use the 2x2 rubric for feedback we discussed earlier.
In the videos, the bash shell for loop is taught, and it is assumed learners are familiar with how to use a variable, the head command and the content of the basilisk.dat unicorn.dat files.
Note: Sometime sounds in the room can be poor. Turning on closed captioning by pressing the cc button will improve the accessibility of these videos.

Watch individually, then as a group use the 2x2 rubric (+ve content +ve delivery, improve content, improve delivery)
Video 1: https://youtu.be/bXxBeNkKmJE
Video 2: https://youtu.be/SkPmwe_WjeY

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. (For this exercise, your peers will not “code-along”, but will instead observe and give feedback.)

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 this exercise. Give each other feedback using the 2x2 rubric we discussed previously and enter the feedback you received in the Etherpad.

This exercise should take about 25 minutes. (back 15:55)

Olivia: Annabel (intro to the command line, pwd, etc): very good pacing, and I like how she explained what the name of the commands mean, so that it's easier for learners to remember. The description of the environment (shell) is a very good thing, not only to explain the commands but she also showed how it affects the shell. My one suggestion is when using a keyboard shortcut, to explain what you are doing, e.g. when using tab to autocomplete. It's hard sometimes because we do it without thinking but it might confuse the learners.
Annabel: Olivia (intro to dplyr). Really good awareness of the R studio layout and indicating which panel to be looking at throughout the coding. Content was very clearly presented and reinforced content of earlier lessons. Perhaps could consciously slow down the pace of delivery (more pauses too) We discussed using analogies and when to step away from terminal, using anecdotes and analogy and how introducing errors might be helpful.

Brian: (Swabir)- Loops! Swabir taught the first stage of the Loops in Python. He paced the lesson well, and demonstrated why doing everything manually was cumbersome. Confident and calm presentation. Hard in 3 minutes, but possibly worth seeking to engage the students a bit more.
Swabir:(Leticia - Loops in Python) - well paced and clear explanation of the actions and output when changing the input in a python loop.
Leticia(Brian): + good context around examples and comfortable relaxed delivery, good at asking for learner feedback to make sure we were engaged
- use of some words that may not be considered inclusive
Lara: - take some breaks between concepts; + engaging, clear explanations; - make sure to not assume any knowledge (see python console/script)
Kiera: Went a little fast, but otherwise good
Paul: I taught at a good pace, however the topic or eppisode part was not a good choice for a demonstration. I should try to have a backup when I may have technical difficulties. I needed to familiarise myself with the demo session. Also my participants could not see my share screen (which froze)

Patricia/Upendra/Kim: Patricia taught how to remove a row in R dataframe, she did an excellent job and was very clear in explaining the logic and how R understands statements. Upendra taught how to print statements in Python and did an excellent job as well, it was good to intentionally make mistakes and explain how errors might arise that way. I taught Python and it was good that I reviewed my content repeatedly after executing the statement

XII. Preparing to Teach
https://carpentries.github.io/instructor-training/15-lesson-study/index.html

Read Software Carpentry’s learner profiles. Note that these example profiles contain more information than you will ever know about a learner; this is a creative exercise in imagining (and empathizing with) the whole people behind the faces. Now, sketch out a profile of someone you might expect to attend your first workshop. Who are they, what problems do they face, and how might this training help them? Be as specific as possible.
Enter your learner profile into the Etherpad.
This exercise should take about 10 minutes.

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

Finish by 4:25 PM NzT

Olivia: Dee Plire, PhD student in microbiology. She has a background in biology, does a lot of lab work, but for her PhD she needs to do some data analysis as well. Excel is not enough and her supervisors are telling her to learn R. She has very little knowledge about programming and finds it a bit scary. She wants to be able to carry out the statistical analyses she needs, and make plots for publications and presentations. If we go too fast during the workshop she will be discouraged and will not go back to R. We need to show her the basics but specially how to learn new things in R, how not to be discouraged by mistakes and how to seek help. The workshop also needs to make here realise that she won't become fluent in R in a few days, but it's totally ok and it's by practising and failing that she will get to the level

she is aiming for.
Kiera: Danny Data: PhD student who has a mixture of data that they've gathered themself, plus a lot of legacy data that they need to tidy up and combine into one nice dataset. They have been primarily using Excel for this, but are frustrated with how long it's taking and the fact that Excel doesn't have a lot of the functionality that they are looking for. They have heard that Open Refine might be a good tool, but are reluctant to throw away all the hard work they've already done with the data files.
Paul: Lecturing Larry: Larry has a PhD and is a Lecturer in economics. Larry has been using statistical software to undertake analyses for years but wants to learn a new open source programming language R/python. He is an expert in the other software language but a novice in R. Larry has a full workload and is quite stressed. He wants to learn R/python as soon as possible to undertake meta-analyses of data collected from government sources. The data contains lots of problems comments and different class entries which needs cleaning before he can undertake an analysis.

Swabir: Eric R- a new PhD student who is working with data from social media that their supervisor has asked them to analyze using R. They're new to coding and data management. The Carpentries will be their introduction to R, and concepts such as webscraping and once they start building their codes, version control and reproducible results.
Patricia: Our typical learner, Bob, is a graduate student with a lot of expertise in their area but minimal coding experience. They have probably collected a ton of numerical data and have been told by their supervisor to go analyse is using this programming language R, about which they know very little. Our learner probably has a life science background and may be quite suspicious of the whole computer thing, so will need to be supported if they start to feel too stressed by the unfamiliarity of the material. In spite of this, they will be interested in high level explanations because they will want to be able to extend what they see in the workshop to their own novel problems.
Oh
Annabel: Petra Dish, PhD student, who has encountered some bioinformatics in their previous learning but not quite within the scope they need for their own project (and with some bad experiences to boot). Dabbled in Codeacademy but nothing really stuck and didn't immediately translate to own project. Skilled in ecology but new to population genomics. They are not surrounded by local support/expertise and have a sense of what they want to be doing with their dataset but not what skills they need to master to get there nor what questions/who to ask.
Zoe Yang: Technicians who work in the labs will start to use whole genome sequecing to study the microorgansism. They may have a little bit of experience of using linux systerm, but most of them are not farmilar with the command line systerm and don't know how to use tools under linux.
Aurelie : PhD student who is asked to process some sequencing data. She is scared of linux and command lines but ready to give it a go. not a lot of experience with computers. she knows it will be useful for a PhD and her carrer
Brian: Reluctant Ron- Ron is a metadata librarian. He's been in the job for decades, and, with only 10 years to retirement, he's keen to Kick Back and Keep Cataloguing until he gets that sweet retirement gold watch. Ron's role is changing though, and if he was able to work with the data a little better, he could hugely diversify the kinds of contributions he can make to the university. Ron is attending because his manager told him to. He's nervous about his ability to learn new tricks, and doesn't want to be put on the spot. Though he is sceptical of the value of the training, he wants to get his boss off his back. If he does learn something new that's of real use, he'd secretly love to be able to teach his colleagues about it and enjoy the delicious library street cred. Ron's near-sighted, and doesn't like to sit around all day.

Upendra: Mike is a first year PhD student in genetics. He has basic programming skills.
He is working on differential gene expression analysis using RNA-seq. It includes QC of raw sequencing data, denovo assembly, quantifying transcripts, differential expression analysis, Go and kegg annotations

and pathway analysis. He tries to find and use GUI based softwares to work through his pipeline and for most of the steps those are not available and if available they are time consuming to process through huge amount of data.
This lesson will show him simple way to import his data in HPC (Nesi), checking md5 for his raw files and a simple commonly used softwares for QC of raw reads and transcriptome assembly. It will demonstrate how to use softwares like fastqc through command lines more efficiently, using softwares like trimmomatic for quality control the raw data and softwares like trinity to denovo assembly the transcript.

Leticia:
Ms Lola Scifan: Lola has just started her masters degree with a research group at the uni she did her undergrad in. She's bright and enthusiastic and loves to interact with her peers. She has good wet lab skills but little bioinformatics experience, other than what she was breifly shown in undergrad.
Her supervisor has unexpectedly changed her project a little, so now Lola has a large next gen sequencing data set to analyse, from scratch. Where to start?!
A genomics data carpentry workshop will show Lola the basics of how to start a project based on a large genomics dataset. It'll show her some best practice ways to go about her analyses and she'll gain an general understanding of the building blocks of working with this kind of data. She'll have her own data to go off and practice on once the workshop is done, and she'll be ready to tackle it with her usual enthusiasm.
Kim: Gerry Saltzmann is a library technician who has worked in the Wen School for 25 years. Early in his career he moved around to different departments around the library but finally landed in the Digital Collections department and has happily resided there for over 15 years. Gerry works on a daily basis in the library catalogue where one day he's cataloguing obscure texts in German, English and French, the next day he's creating controlled vocabulary lists for music in the Roman Empire during the Medieval period. Gerry is finding that often times he has to go in and update records because the system constantly is being updated by the vendor and is changing around how the underlying data is being modelled and represented in the catalogue. He knows that the catalogue has an API and he sees that the vendor always gives instructions along with sample scripts in Python on how to use it for every software update, but he doesn't have any experience with programming and doesn't know where to start.
Kari: trainee intern tina has 300 patients who have had colorectal surgery. she is recording some post surgical outcomes, and integrating patient demographics from the electronic patient management system. she has no coding experience, limited statistical knowledge, and works a milion hours a week on call. She would like to analyse her data and look at the associations between patient frailty and outcomes.

Feedback end of day 3: https://forms.gle/pLrVLADc7pvZkqbz7

Day 4

XII. Preparing to Teach cont.
https://carpentries.github.io/instructor-training/15-lesson-study/index.html

Reverse Instructional Design (and Preparation!)

Working With Learning Objectives

Evaluate Carpentries Learning Objectives (10 min) (end at 13:00)
Select one learning objective from one of the lessons linked to below and then complete the following steps to evaluate it.

- https://librarycarpentry.org/lessons/
    - e.g. https://librarycarpentry.org/lc-open-refine/04-faceting-and-filtering/index.html
- https://datacarpentry.org/lessons/
    - e.g. https://datacarpentry.org/spreadsheet-ecology-lesson/03-dates-as-data/index.html
- https://software-carpentry.org/lessons/
    - e.g. http://swcarpentry.github.io/shell-novice/02-filedir/index.html
    - e.g. https://swcarpentry.github.io/python-novice-inflammation/02-numpy/index.html

- Identify the learning objective verb. How specifically does this verb describe the desired learner outcome?
- Where does this verb fit on Bloom’s taxonomy? Do you think this is an appropriate level for your learners?
- In your opinion, does the lesson do an effective job of meeting the stated objective?
- What would the next level on Bloom’s taxonomy look like for your learners? How might you be able to help them think ahead to the next level without attempting to get them there during your workshop?

Blooms taxonomy diagram: https://carpentries.github.io/instructor-training/fig/Blooms.png

Olivia: Creating Publication-Quality Graphics with ggplot2 (http://swcarpentry.github.io/r-novice-gapminder/08-plot-ggplot2/index.html) - Verbs: "be able to use", "apply", "manipulate", "improve", "save": the first one is a bit vague, the others more specific and focus on some aspects of the problem (here creating graphics with ggplot2). "Be able to use": create; "apply" : apply, "manipulate": apply; "improve": create
Kari: Short intro to python (https://datacarpentry.org/python-ecology-lesson/01-short-introduction-to-Python/index.html), fits on the "remember" and "understand" rungs of the diagram. This is definitely appropriate for beginners, and the lesson meets the stated objectives. To encourage next level a teacher could encourage learners to use their own datasets and explore the basic tasks given in the lesson, and think about how they could use it to optimise their time spent in their work.
Kiera: Working with Open Refine (https://datacarpentry.org/OpenRefine-ecology-lesson/01-working-with-openrefine/index.html). THere are several objectives here, and they fit under the create, apply and remember categories. I'm not 100% sure the recall ones are appropriate because I don't think we've covered what facets are yet, so unless they knew that information previously there's nothing really there for them to recall. Other than that, the objectives are appropriate and the learners definitely meet the objectives by the end of the lesson. Next level for my learners might be 'understand' - so figuring out where and when they would use this within their own work.
Zoe: Navigating the filesystem(https://librarycarpentry.org/lc-shell/02-navigating-the-filesystem/index.html)
fits on the remember, understand and apply of the taxonomy. it suits for the beginners and it meets the stated objectives. The next level of my learner might be start working on the their data in linux systerm using comannd lines.
Swabir: Analyzing patient data (https://swcarpentry.github.io/python-novice-inflammation/02-numpy/index.html) - This would be in the first, second and the third levels. The actions are to explain, which

require one to reall some facts. Then there's the objective of being able to apply some previously learned actions in new situations (import, read ...)
Upendra: Navigating files and directories (http://swcarpentry.github.io/shell-novice/02-filedir/index.html)
I think the objectives fits under understand and apply in blooms taxonomy. for a data carpentary workshops where most of the learners are beginners it is an appropriate level.
the next level would be to use the understandings for using soft and hard links for the path

Kim: https://librarycarpentry.org/lc-shell/04-loops/index.html
- Build a concept for how loops can be used to repeat tasks
- Implement a loop to rename several files
- understand - "building a concept not create"
- yes appropriate level
- yes meets objective, introduces loops, uses an example human readable, then uses example in bash building upon knowledge
- next level, apply is to do an exercise, maybe get them there by talking about how you can do this in their real life situations brainstorm what files they have that they could do stuff to
- apply - "implement"
- yes appropriate
- yes meets objectives
- next level is to analyze, which an example prompt could be to ask learners to compare doing this manually or when they had to do that vs how you might use this to automate
Paul: SQL - The learning objectives follow blooms taxonomy, First introduce the topic and cover the basic functions needed to creat and interact with SQL. It reinforces the knowledge and understanding by doing coding and formative assessments . Explain what is SQL and how to use/apply it. ‘Create’ Identify library and informative skills that relate to SQL.

Let's finish it by 1:20 PM NZT

Have a look at your lesson again. Choose a learning objective, and identify *where* in the lesson that objective should reasonably be achieved. How will you know that that objective has been met for all learners? Will this be clear to them?
Make a plan for *where* in your lesson you will use different types of formative assessment to help everyone in the room monitor their progress. Keep in mind that formative assessment can take many forms, including multiple choice questions, faded examples, spontaneous questions and calls for sticky notes. Write some notes or thoughts about this process in the Etherpad for discussion.
This exercise and discussion should take about 10 minutes.
Kari: beginning in Python: objective: define data types: I would start be asking for sticky notes to see if anyone has a question, and then give them faded examples of each where they either provide the definition or an example.
Olivia: Creating Publication-Quality Graphics with ggplot2. Learning objective: "To be able to use ggplot2 to generate publication quality graphics." -> faded example throughout the lesson, at the beginning only 1 or 2 blanks, by the end of the lesson they have to fill almost everything
Lara: Handling data with Python. Learning objective: "Access data from dataframes". Monitor: Ask for a value in a specific column and row of the dataframe (requires loading the dataframe, understanding structure, manipulate dataframe)
Kim: - "Build a concept for how loops can be used to repeat tasks"
- I would ask everyone to write a summary of a loop with real life objects
- to talk about the previous state (some dogs are stray and happy and free, some dogs have collars because

they have owners)
- then talk about the loop: for each stray dog in the dog park , put a collar on it
- then talk about what would the end state be? (all dogs would have a collar)

Zoe:Navigating the filesystem. I will ask them a question"which directory you are currently work on and list the files in that directory."

Brian: OpenRefine: Facets and filtering. "Use facets and filters to work with a subset of data." I'd include a simple example with provided data. I'd try to set the data up in such a way that I can ask a question like "How many rows are there with some variation of the name Brian/Bryan." The idea is that if we make a mistake, I'll be able to use the result to diagnose where they're going wrong.

Upendra: Navigating the file system: ask them to give a absolute and relative path for a file in another directory.

Patricia: "Understand" what a factor (categorical variable) in R is. Set up vector of a factor data type with multiple instances of different levels. Have learners predict result of levels() command. If they don't understand what a factor is, they won't translate from the multiple instance vector to the unique instance factor.

Paul: Data Carpentry - learning objective - Indexing and subsetting data frames: Formative assessment ; ask them to filter a provided dataset and respond to what the filtered dataset contains.; Ask them what would you write to filter the dataset by *x* variable; how do I filter a dataset to remove a variable or entry from the dataset

Leticia:'Explain the similarities and differences between a file and a directory.' could be good to get them to take an adventure to various directories and list contents, keeping track of how many directories and files are in each. They'll need to give total at the end.

Swabir: Analyzing Patient Data - Read Tabular Data - Check that they can identify the different parts of the calls used to load the data into a variable. Used progressive blank spaces in the commands to make sure they recall and can explain what each piece of the lines do.

Kiera: Working with Open Refine - Get students to tidy up the 'scientfic name' column of data by combining similiar names together. See if they can get to the correct number of total names

XIII. More Practice Live Coding

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

Round Two (25 min) finish 13:53

First, have a look at the rubric that is given to trainers as a suggested framework for evaluating the online teaching demonstration sessions that are part of instructor checkout: https://carpentries.github.io/instructor-training/demos_rubric/. Does this rubric make sense? Take a moment to think about things you'd like to do differently with your next live coding practice. This is also a good time to ask questions about teaching demonstration.

Next, 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 and any new ideas based on reading the rubric. Give feedback to each other using the rubric this time.

When you are finished, add some thoughts on this process to the Etherpad: What did you change? Did it work better or worse with the change? How might you do it if you were to teach it again?

Key Points

- (Reflective) Practice makes perfect.

Olivia: Annabel went through the intro to command line again. She used the feedback from yesterday to improve even further her live coding. It was excellent, she used an analogy to explain the concept of folders and navigating through the file system. Also explaining what each command does and why/when we want to use it before demonstrating is very good, then students know why we're learning that.

Not exactly about change, but we did note that Python occasionally throws out peculiar roundings (0.1 + 0.2 = 0.3000000004), arguing for the value of a thorough practice of your material before you take it in front of a classroom
found it better today, was much more prepared. Three minutes goes by quickly!
Swabir: An important point we discussed - It's better to pace yourself while delivering the material, even if you don't get through it al, than rush through when you're short on time.
- There's always more you can explain, it's a good idea to anticipate potential questions as you're going through your lesson and if you can think of the potential questions, explain it
Leticia: much more relaxed today and it was noticable in how i presented the material. Better for all parties involved.
Kiera: Tried to slow down and added a real life example to make the lesson more applicable to real life
Brian: My pacing was better, I think I was better at viewing what I was doing from a learner perspective.
Annabel: I think we both improved our delivery in light of the feedback yesterday and our reflection on this. One material impact of that was that we covered far less material.
Paul: Neg: I probably should have explained the names of the functions I was using at length more, and before using them. I should have taken a extra moment of preparation before commencing; Pos - There was a good pace and presented the content clearly

Welcome back to second part of day 4!
<u>* How do you actually start a workshop?</u>
Create an outline of important points to cover in an introduction. Try to group points that belong in the same category.
(6 minutes)
Categories

1. Set positive first impressions
2. Introduce yourself effectively (and have other workshop leaders do the same)
3. Clarify learning objectives and expectations
4. Help learners learn about each other
5. Set the tone for the workshop
6. Collect baseline data on learners’ knowledge and motivation
7. Whet learners’ appetite for workshop content
8. Inform Learners of Logistics

Code of conduct3
Introduce what you hope to cover clearly. 3
Talk about what's in it for them, maybe invite information about what they hope to get out of it (a chance to establish partners in crime maybe?) 74
Check that everyone's screens look more or less the same as yours, if that's how it should look. 8

Health and safety8
Mention the version of the software you will be using and perhaps to expect different results if they are working witha different version. tell them to discuss with the person next to them or a helper if they get something different.
Background of myself and helper, what's the main purpose for the workshop. what we need before start. Describe the main features of the specific softwares. Make sure the learner all have it install. 2
Which software do you need, if someone doesn't have it, take a few moments to do that or a helper can help
Link to the data carpentry lesson so that if someone feels they are lost they can come back to that. 7. Mention the Carpentries code of conduct. 5. When doing the introduction of each learner, ask them to tell why they registered to this workshop 4, 7
Ask a few people why they want to learn ____ in the session today (what are thier assumptions of the learning objectives), and if other participants have the same learning objectives 3
A good runthrough of where to find the worshop material on the Carpentries webstie, and how to follow along through the episodes of a lesson. Often overlooked, but the layout can be confusing. 8
Clearly state your assumptions about prior knowledge so they're not afraid that they are underqualified
Emphasise like you mean it that all questions are ok3
Make sure everyone has access to the resources/links they need before starting - 8
Get learners to talk from the very beginning - it'll encourage them to keep talking and participating - 5
Personal introductions (self and co-trainer/helpers) with emphasis on approachability and positive classroom environment2, 5; discuss motivations for running/attending the course, broad-brush objectives and some of the skills you'll be introducting; how to ask for help (code of conduct, sticky notes etc); logistics of the venue and the schedule.1
Introduce myself2, get the participants to briefly introduce themselves4, what they're studying and what they want out of the course3-7. Explain the purpose of the course and the rough outline of what we'll be doing. Make sure everyone has the appropriate software, let them know that they can ask questions, get help at any time. 2, 3, 4
Introduce myself and helpers (who we are, what we do, why we want to teach the wkshp) 2
Say what the subject is, objectives, and timeline 3, 8
Give course details online/etherpad details 8
Ask others to introduce themselves 2
<icebreaker of some sort> 4
Ask what they would like to achieve 7
Go over the objectives and timeline again
Discuss logistics (exits and/or zoom stuff)
introduce yourself and your background, introduce the helpers, go over the material you hope to cover and expected timelines, take a poll of the participants backgrounds and expectations, give overview of the (coding) environment as well
Welcome everyone and introduce yourself, supporting instructors and helpers.2
Clearly mention what your assumptions are about the participants knowledge. 6
what you are going to cover and the possible outcomes 3
make them comfortabl to ask questions and make sure everyone has all the required softwares to start with

XIV. Managing a Diverse Classroom

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

What Are the Challenges? (10 min) back 15:27

What are the challenges for running is workshop (Instructor Training) given how diverse the audience is?

Breakout room 1: Class is online vs. in person, timezones, access to internet that's appropriate for the Zoom sessions, changing the session from a two day workshop to a four day, being online can be distracting, people coming from different backgrounds with different experiences --> might need different levels of teaching. Not having people in front of you to give the visual feedback you'd normally get, esp. when sharing screen

Breakout Room 2: People come from different backgrounds and timezones, and their internet connections might not be robust enough for them to participate fully. Some will have other barriers to participation, even things like shyness. A variety of technical expertise could be an issue.

Room 4: We as learners have different levels of teaching experience, and one of the challenges is being able to teach each of us where we are at in our skill level/understanding (both in teaching experience and in coding). We also discussed the timezone differences and technical difficulties.

Room 3: We are concerned about how to handle a situation where a small number of learners are struggling while every else has finished. Suggestions: assign a helper, provide additional exercises for the faster workers to practice when they are done. If there are only a few really fast ones, you can quickly come up with some extra things for them to work on.

We acknowledge that this particular workshop must be challenging because of the wide range of prior teaching experience, but not too difficult to cope with the professional area differences, because the skills being taught are primarily discipline agnostic.

The Carpentries Handbook, all the info about The Carpentries
https://docs.carpentries.org/

Code of Conduct: https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

Alternative:
https://docs.google.com/forms/d/e/1FAIpQLSdi0wbplgdydl_6rkVtBIVWbb9YNOHQP_XaANDClmVNu0zs-w/viewform

Teaching Together - Nuts and Bolts back at 15:53
With a partner, imagine that you are planning a workshop together and answer the following questions:
* Which parts and how you divide between temselves?

- How would you prepare to teach a workshop together?
- What would you like to swap maybe? Try negotiate :-)

Picking up the Pieces 5mins (back 16:05)

In small groups or on your own, make a list of all the concepts you've encountered in this training. Your list can include everything from educational/teaching theories to specific in-classroom practices.
Bloom's taxonomy & threshold concepts
novice vs experts - how we think differently and make connections more easily as experts
Choosing your language carefully to create a supportive and constructive learning environment
Formatiive assessments- all the time! Use them with feedback for learning!
Room 2: formative vs. summative; novice/competent/expert -- awareness gap; mental models; cognitive load; chunking; motivation and demotivation; effective feedback processes content/presentation quadrant; taxonomy of learning objectives; live-coding practice (SLOW DOWN!!); being ok with slowing down -- it's more important that they learn than that you reach the ends of your notes; great to join community; we're all knackered.

Questions remaining:

- when will the NZ checkout discussion sessions be up for signup? not teaching demo, the discussion
    - line 231 https://pad.carpentries.org/teaching-demos
    - https://pad.carpentries.org/community-discussions

https://carpentries-incubator.github.io/instructor-training-bonus-modules/

Post Workshop Surveys (5 min)
Assessment is very important to us! Please complete this five-minute post-workshop survey. (https://www.surveymonkey.com/r/instructor_training_post_survey?workshop_id=2020-10-14-ttt-online-nz)

Key Points

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