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](https://etherpad.wikimedia.org/)).

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

All content is publicly available under the Creative Commons Attribution License: <https://creativecommons.org/licenses/by/4.0/>

 ----------------------------------------------------------------------------

**FAIR in (Biological) Practice**

<https://edcarp.github.io/2022-02-15_ed-dash_fair-bio-practice/>

**Day 3**

(Day 4 notebook <https://pad.carpentries.org/2022-02-18-ed-dash-fair>)

**List of attendees**

1. Kadi Vaher
2. Nneka Nnadi
3. Amelia Edmondson-Stait

4. Flávia Fonseca Pezzini

 5. Caity Ellis

 6. Wu Huang

 7. Cigdem Selli

8. Adelaide Young

 9. Ines Boehm

 10. T Zhou

11. Marta Amador

12. Alex Meier

Sumy Vnnilathil Baby

Your coding experience,

Please, enter +1 if:

0: Not much coding experience: +1 (i have used R once)

1. You use jupyter notebooks:+1(sometimes) +1 (once)+1(sometimes)+1(several)+1+1(sometimes)+1

2. You sometimes write scripts (bash etc):+1+1+1+1+1+1(sometimes)+1+1+1

3. You sometimes code in R: +1+1+1+1+1+1(often)+1+1+1+1+1

4. You sometimes code in Python:+1+1+1+1(sometimes)+1+1+1

4. You code in other language (type which):

 I am being kickout from the zoom

 do you want to try restart your internet?

------------------------------------------------------------------------------------------------------------------

**Lesson 10: Jupyter notebooks for reusable data analysis Exercise 1: Basics of Jupyter Notebooks**

Open this Jupyter Notebook: <http://mango.bio.ed.ac.uk/jupyter>

We will first show you how to duplicate a notebook, save it and run code.

1.     Select the notebook titled **'student\_notebook\_light\_conditions.ipynb**' and click 'Duplicate'.

2.    Confirm with **Duplicate** when asked if you are certain that you want to duplicate the notebook.

3.    A copy of the notebook has appeared with the suffix '-Copy' and a number. Explore the anatomy of the notebook (<https://github.com/carpentries-incubator/fair-bio-practice/blob/gh-pages/fig/10-02-jupyter_anatomy.png>)

4.    Change the title of the notebook from -copy number to your initials e.g. “student\_notebook\_light\_conditions\_IB”

5.    Save the notebook: Click on the disk symbol in the toolbar

6.    Run the notebook: Select the top cell of the notebook with the title (this is likely pre-selected already and will show with a light-blue bar to its left), and click “Run” in the tool bar.

7.    Click two times. What can you see?

-

-

-

-

8.    We want to run ALL the code: In the top tool bar click Cell > Run All. What can you see?

-

-

-

-

 DONE: +1+1+1+1-+1+1+1+1+1

 ------------------------------------------------------------------------------------------------------------------

**Exercise 2: How to add and remove content**

You have duplicated a notebook and saved it under your own name. Now we will add text, remove cells and change code.

1.     Change the author name of the document to your name: Double click on the cell containing the author name and change the name.

2.     Press Run again.

3.     Add a new cell: Let us add details about the “light\_results.txt” file that is loaded. The “+” in the tool bar creates new cells below the currently selected cell. Thus select the cell above the code and click “+” in the toolbar.

4.     Ensure the type of the cell is Markdown and enter a description of subsequent analysis e.g.: “Loading of results following short- and long-day light exposure on arabidopsis, followed by visualisation of differences in chlorophyll/biomas etc... content between genotypes on short-days and long-days.”

5.     Press Run again.

6.     To remove a cell, select the cell you have just created and click on the scissors icon in the toolbar. (This can be undone under Edit > Undo Delete Cells)

7.     Change colours of your graph: Where the code of the graph reads the comment “# change colour of groups” you can replace the HEX codes, # followed by 6-symbol code, with names of colours (e.g. blue, green…) or other HEX codes if you are familiar with them.

8.     Save graph under new name: Add your initials to the file name under which the image is saved. Press Run. Your image should be visible in overall file hierarchy.

DONE: +1+1+1+1+1+1+1

 ------------------------------------------------------------------------------------------------------------------

**Exercise 3: Add another analysis step**

We have shown you how to manipulate text and code in Jupyter notebooks. You should be able to add data visualisation (a graph) and stats for long-day light condition including annotations yourself.

1.     Add additional cells including

a.     Titles

b.     Edited code to depict graph from long-days (saved under different name)

c.     Figure legend

d.     Statistical testing of difference between genotypes on long-days (remember to assign a different variable throughout e.g. LD.aov)

e.     Interpretation of results of statistical testing

DONE: +1+1+1+1+1+1

 ------------------------------------------------------------------------------------------------------------------

**Exercise 4: Sharing of your Jupyter Notebook**

You have now generated your own analysis and interpretation on top of your collaborators results and want to share this with your colleagues.

1.     Download your Notebook (ensure all code has been run) as .html and .pdf

2.    View the documents and think about why it is important to run all code before download (try Cell > All Output > Clear and download your Notebook, compare the outputs)

**Questions:**

What is the difference between running all code and clearing all run code?

Why is it important to run all code before download?

 Download file is static, doesn't allow for execution. Therefore only previsouly calculated output can be viewed.

DONE: +1+1+1+1

 ------------------------------------------------------------------------------------------------------------------

 BEING BACK 14:10

**Exercise For Advanced**

FInd a way to plot in sensible way: both conditions LD and SD on the same graph, for the 3 genotypes and two output variables (biomas, starch).

If you created advanced plots type your name bellow:

 Kadi (but I didn't put the two output variables onto the same graph but rather did one fig for biomas and the other for starch)

You can try to create the same plots in python (but we cannot guarantee all the packages are installed)

 ------------------------------------------------------------------------------------------------------------------

**Exercise 5: Accessibility of Jupyter Notebooks**

On a scale from -2 to 2, how do you feel about the following statements (R is interchangeable with Python),

where -2 (strongly disagree), 0 no opinion to +2 strongly agree:

Type a number between -2 to 2 behind the statement:

·       making graphs for a subset of data is easier in R than in Excel:0+2+2+2-1+20

·       it is easier to filter for data in R than in Excel: +2+2+2+2+2+1+2

·       it is easier to generate a series of plots with similar layout in R than Excel:+2 +2+2+2+2+2+2

·       it is easier to do large scale data processing in R than in Excel: +2+2+2+2+2+2+2

·       using notebooks does not require any programming knowledge:+1+1-100-1-1

·       notebooks give you a better overview of your data analysis than Excel: +2+2+2+2+2+2+2

·       notebooks links laboratory style records with data analysis:+2+2+2+2+2+2+2

·       Jupyter is free, whilst a Microsoft Office (+Excel) suite costs $149.99, this alone is an incentive to use Jupyter: +2(but this applies in general for using R/Python over Office not Jupyter notebooks specifically)+2people also use open office +2

·       you need to learn R to do any data processing in notebooks:-1 -2(when rerunning someone else's code) +1+1-10-1

·       notebooks assures reproducible computing:+2+2+2+2+2+2+1(depends on documentation of the person who generated the notebook)

·       wrong inputs or not captures parameters are main reasons for not reproducible analysis:+10+2+10+10

DONE:+1+1+1+1+1

------------------------------------------------------------------------------------------------------------------

Back 2:50

**Lesson 11: Version Control**

**Exercise 1: Problems with a change 2:53**

Which of these issues can you relate to? (Type +1 next to each statement if you relate to it)

·       I have fifteen versions of this file - which one do I use? +1+1+1+1

·       I can’t remake this figure from last year.+1+1

·       I slightly modified my code in one place, everything stopped working. +1+1+1+1+1+1

·       I have several copies of the same directory because I'm worried about breaking something.+1+1

·       Somebody added erroneous records in a shared file with samples, cannot find who and why+1

·       You remember seeing a data file but cannot find it anymore: is it deleted? Moved away?+1+1+1+1+1

·       I tried multiple analysis and I don't remember which one I chose to generate my output data+1

·       I have to merge changes to our manuscript from 10 different emails with collaborators+1 +1+1+1+1+1

·       I accidentally deleted a part of my work+1+1+1

·       I came to an old project and forgot where I left/saved it+1+1+1

·       I reorganized my data files, but broke the analysis pipeline and don't know how to fix it+1+1

·       I have trouble to find the source of a mistake in an experiment+1

·       My directory is polluted with a lot of unused/temporary/old folders because I'm afraid of losing something important+1+1+1+1+1+1000+1+1

·       I made a lot of changes to my paper but want to bring back one paragraph+1+1+1+1

DONE:+1+1+1+1+1+1+1

 ------------------------------------------------------------------------------------------------------------------

**Exercise 2: Manual Versioning**

Which of these issues does manual versioning help to tackle?

(Type +1 next to each statement if manual versioning helps to tackle)

·       I have fifteen versions of this file and I don't know which one to use +1+1+1+1+1

·       I can't remake this figure from last year+1+1

·       I slightly modified my code in one place, everything stopped working +1+1+1+1

·       I have several copies of the same directory because I'm worried about breaking something+1

·       Somebody added erroneous records in a shared file with samples, I cannot find who and why +1

·       I tried multiple analysis and I don't remember which one I chose to generate my output data +1+1+1+1

·       I have to merge changes to our manuscript from 10 different emails from collaborators+1

·       I made a lot of changes to my paper but want to bring back one paragraph +1+1+1+1+1+1+1

 Mac automatic backup failed and the file just disappeared - icloud!

 Also painful if you run out of space on your external hard drive for your time machine backups...

 Do people use git with dropbox?+1 in case folders are already backup in dropbox. Ok thanks

DONE:+1+1+1

 ------------------------------------------------------------------------------------------------------------------

**Exercise 3: Changelog in action**

Have a look at one of the example Github repositories and how they track changes\*:

Green and Blue:

·       data from E.R. Ballou et al. 2020

·       <https://github.com/ewallace/pseudonuclease_evolution_2020/commits/master>

Give examples of:

·       what makes them a good changelog

* Author and Date are persisted. Individual changes are visible and accessible. Branching/Forking is an option.

·       what could be improved

 - could be more detailed about what is changed in which files

 Commit messages often lacking verbosity (word limit)

Which are the most difficult features to replicate with manual version control?:

 -track the changes made

Red, Yellow:

·       data from I. Boehm et al. 2020

·       <https://github.com/BioRDM/nmj-pig/commits/main>

Give examples of:

·       what makes them a good changelog

 - regular comits

·       what could be improved

 -names are too long

 - too many changes in one file

Which are the most difficult features to replicate with manual version control?

 DONE: +1

 ------------------------------------------------------------------------------------------------------------------

**Exercise 4: 3:36**

Five reasons to use a version control system in research, give +1 to the two most important for you

·       Tell the story: The history of your commit messages will describe your project progress.

·       Travel back in time: a version control system makes it easy to compare different time points of your project smoothly. If you want to compare the stage of your project a year ago from now, it only takes one command-line of code+1+1+1+1+1+1+1

·       Experiment with changes: if you want to make changes in a script you can first make a “snapshot” of the project status before experimenting with changes.+1+1+1+1+1

·       Backup your work: by being able to link your local repository (folder) to a distant online host (GitHub), a version control system backs up your precious work instantly +1 +1+1+1+1+1

·       Collaborate easily on projects: having a web-hosted synchronised version of your project will encourage collaboration with other researchers. Think about a colleague of yours being able to add a script to make a figure for your publication for instance. The repository will track their contribution and evidence their participation in the project +1

DONE:

 ------------------------------------------------------------------------------------------------------------------

**Learning GIT:**

 •Software Carpentry git workshop <https://swcarpentry.github.io/git-novice/>

•Edinburgh Carpentries runs courses: <https://edcarp.github.io/>

•<https://ourcodingclub.github.io/tutorials/git/>

•Crash course on youtube: <https://youtu.be/SWYqp7iY_Tc>

•Learn git branching: <https://learngitbranching.js.org/>

•Learn git-game: <https://github.com/git-game/git-game>

 Back at 4.00

 ------------------------------------------------------------------------------------------------------------------

**Semantic versioning quiz**

Type +1 after the correct answer:

1. Which     of the library version is the latest?

·       a) 0.12.4

·       b) 1.1.0

·       c) 1.12.3-alpha+1 +1+1+1+1+1+1

·       d) 1.12.2

1. You     use library fair.2.3.1 to automatically upload your data after analysis.     You found out that there is available new version of the library     fair.4.3.1. Which situation is the most probable?

·       a) fair.4.3.1 uploads data twice as fast as fair.2.3.1

·       b) you need to reserve time to test the new version and adapt your analysis pipeline to work with it +1+1+1+1+1+1+1

·       c) both versions offer the same features and bug fixes as they have the same minor/path version 3.1

1. You     developed code that helps to plot detrended and normalized data, the last     release has version 1.2.1. You added Z-score to the available normalization     methods and fixed the spelling errors in the user messages. Which version     number should you give to the new release:

·       a) 2.0.0

·       b) 1.3.0!

·       c) 1.3.2

·       d) 1.2.2+1+1+1+1+1+1

·       e) 2.2.1

1. Your     group releases data about studied cancer patients following semantic     versioning. The last data sets are released as 1.11, you added data from a     new patient,
2. what     version should you use:

·       a) 1.12 +1+1+1+1+1+1

·       b) 2.11

·       c) 2.0

1. You     were asked to better anonymize the above cancer data (v1.11) in future     releases. Instead of data and time of patient visit you release only the     year, the hospital names have been encoded and patient age has been     obscured by random noise.
2. What     version should you use:

·       a) 1.12

·       b) 2.1

·       c) 2.0+1 +1+1+1+1+1

DONE:

------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------

**Lesson 4: Intellectual property, Licencing and Openness**

**Exercise 1. Checking common licence: 4:30**

1. Open CC BY licence summary <https://creativecommons.org/licenses/by/4.0/>

is it clear how you can use the data under this licence and why it is popular in academia? +1+1+1+1+1+1

2. Check the MIT licence wording: <https://opensource.org/licenses/MIT>

is it clear what you can do with software code under this licence? +1+1+1+1+1+1+1

3. Compare the full wording of CC BY

<https://creativecommons.org/licenses/by/4.0/legalcode>

can you guess why the MIT licence is currently the most popular for open source code? You don't have to ask for permission to adapt it? Ha I didn't read all of it in detail - it shows!

DONE:

**Lesson 12: Templates for consistencyExercise 1: Your template**

A text like example:

<https://github.com/carpentries-incubator/fair-bio-practice/blob/gh-pages/files/readme_template_BioRDM.txt>

**Excercise 1:**

In your group, decide what should be included for either

- a measurement/assay (for example: PCR on robot)

- experiment (for example: gene levels in response to stress)

and write your own template

As Excel sheet, or a document (txt/word) or in Benchling.

\* provide example inputs (especially for multi values)

Folder to drop templates:

<https://uoe-my.sharepoint.com/:f:/g/personal/tzielins_ed_ac_uk/EtzDrquhA45KlqFPSP3HFL0BpfbwW5ANJKdV00gRqLI2kA?e=QIwXON>

Exchange emails in case the zoom dies!

-----------------------------------------------------------------------------------------------------------------

**Q&A:**

Do you have any questions about the topics dicussed today? Please write them down here. Use +1 to upvote the ones you are interested in if someone already asked it. We will briefly discuss them before the following set of lessons.

-

-

-----------------------------------------------------------------------------------------------------------------

**Feedback:**

1.      How do you feel about the presented topics after this session (type +1 next to the statement that best describes your feeling):

•       I am more confused:

•       I have a better understanding of them now: +1

•       My knowledge has not changed much:

2.      How was the pace of the lesson:

•       Too fast:

•       About right:+1

•       Too slow:

3. From 0 to 5 how useful was the jupyter demo: 5

4. If the lesson could be 5 minutes longer, what would you add or spend more time on:

-

5. What could be improved:

-

-

-

6. What did you like: