



Data  
Schools

# The Carpentries: Teaching

Bianca Peterson  
North-West University  
[bianca.peterson777@gmail.com](mailto:bianca.peterson777@gmail.com)

# Let's look at:

- Essentials to teach
- Which platform to use?
- Code-along vs presentations?
- What data to use?
- Focus on exercises!
- Materials
- Final thoughts

# Essentials to Teach

- For each module:
  - Layout (OpenRefine, R, Python)
  - Terminology and syntax
  - Structure of commands
  - Absolute vs relative paths (NB!)
  - Where to find help
- Overall: how does each module fit in the “big picture”?

# Which platform to use?

- OpenRefine – GUI (java application)
- Unix – terminal
- Git – terminal
- R – RStudio Desktop / Cloud
- Python – plain interpreter (terminal) / iPython interpreter / Jupyter Notebook

# Code-along vs presentations?

- The Carpentries follow an applied approach - avoid the theoretical (presentations) and incorporate practice (code-along).
- From personal experience (teaching ~4 years):
  - Typing slow – fall behind and/or miss commands
  - Providing code in advance – lose attention
  - Balance: scaffold-script + presentation + code-along + exercises (formative assessments)
  - Bonus: cheat sheets / command summary sheets

# What data to use?

- The Carpentries lessons make use of data that is relatively easy to understand.
  - SWC – domain-independent data (give relevant examples - depending on the group).
  - DC – domain-dependent data (use when teaching learners from a specific domain).
- Each module makes use of a different data set – may add to learners’ cognitive load.
- Whichever data you choose, make sure to explain it before using it.

# Focus on exercises!

- Prioritize exercises - doing exercises is a huge part of the hands-on Carpentries model and also gives people a chance to chat with their neighbors.
- Learners should be able to practice what they are learning in real time, get feedback on what they are doing, and then apply those lessons learned to the next step in the learning process.
- Do simple incremental exercises (start with scaffolds).
- Old-school group activities.
- Space practice over time.

# Materials

OpenRefine: DC - <https://datacarpentry.org/OpenRefine-ecology-lesson/>

Unix: SWC - <http://swcarpentry.github.io/shell-novice>

R: SWC - <http://swcarpentry.github.io/r-novice-inflammation>

Python: <http://swcarpentry.github.io/python-novice-inflammation>

Git: <http://swcarpentry.github.io/git-novice>



# Preparing to teach

- Read through the instructor guide for the lesson.
- Test commands and think about what you need to say (code-along).
- Prepare analogies to explain difficult concepts.
- Use multi-modalities, e.g. speak while showing a graphic (complement).
- Pick exercises.

# Final thoughts

- Diverse classroom.
  - Many people with widely varying skills and experiences.
  - Learner's backgrounds, motivations, and how they learn (educational psychology) – help us to teach more effectively.
- Supportive learning environment.
- You don't have to teach them everything you know.
  - It's better to teach the basics really well, rather than rushing to get through tons of material.
- Teaching is a skill that requires practice and feedback – it improves with practice.



Data  
Schools

Thank you!