Module 3: R

Submodule 2: Work practices

Expected length: .5 day

Guiding question: How can we avoid getting stuck on errors while using R? How does R help us work reproducibly?

Concepts: preventing errors, isolating errors, searching for solutions, reproducible examples

Description: This module teaches students how to approach errors in their code and how to make sure their work is reproducible using RProjects and good coding practices.

Instructor Preparation: Run all code that is supposed to generate an error on your system to ensure it generates the same error as described in the lesson.

| Materials and resources | Learning objectives |
| --- | --- |
| 02-work-practices\_deck.html | 1. Functional problem-solving abilities for learning and using R  2. Use an RProject and GitHub to make your data analysis project reproducible  3. Understand coding conventions |

| Length | Lesson content | Guidelines, tips, and tricks |
| --- | --- | --- |
| 60min | Errors  - Getting help  - Using stack overflow  - Making reproducible examples  - Debugging  (Alexander (eds), 2021 Chapter 10-15) | These will be very abstract concepts unless students have previously worked with code. For the ideas to actually be absorbed, they should be introduced at this point and reiterated as students move through the rest of the submodules. |
| 60min | Formative exercise  1. Create a personal flow chart for debugging errors. | Ideally the students will have something that they can refer to and add to during the rest of the module. |
| 30min | Reproducibility  - RProjects  - Coding practices  (Wickham and Grolemund, 2017 Chapter 8; Alexander (eds), 2021 Chapter 19) | Demonstrate setting up a project live. |
| 60min | Discussion Questions  2. What debugging techniques do you think will be the most helpful?  3. In your own work, what challenges do you anticipate in making processes reproducible? What about in the work of others? Brainstorm ways that these concerns could be addressed. |  |