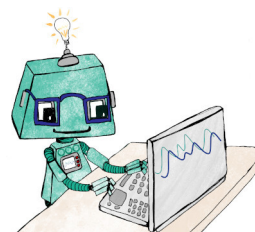


R Best Practices

So you can write code? Great. But can you write code which is easy to read, simple to maintain, and reproducible? Under the pressure of deadlines even the best of us can fall victim to bad-practices. In this course we motivate the importance of good-practices, and show how we can make best practices second nature by incorporating them into our normal workflow.



Course Outline

- **Naming:** Creating user-friendly variable and function names.
- **Writing Modular Code:** Functions that do one thing, well.
- **Documentation:** Writing helpful comments & documentation.
- **Code Style:** Using spacing and conventions to make code more readable.
- **Testing:** Code coverage and error handling.
- **Working Collaboratively:** Techniques such as code reviews and pair programming.
- **Reproducible Workflows:** Introduction to version control and building an analysis workflow.

Learning Outcomes

Session 1:

By the end of session 1 participants will...

- understand the importance of clean code and using meaningful names.
- be introduced to various naming conventions.
- be familiar with the `{janitor}` package.
- explore the value of a consistent code-style.
- write functions to minimise duplication.

Session 2:

By the end of session 2 participants will...

- reflect on and refactor code.
- catch and fix simple mistakes in formatting with `{lintr}` and `{styler}`.
- write code for humans, not computers.
- understand the levels of reproducibility.
- work collaboratively through code review and paired programming.

This course does not include:

- Much detail on using the `{tidyverse}` to manipulate and clean data ready to be plotted, for this see the [Data Wrangling with Tidyverse](#) course run by Jumping Rivers.

Attendee Feedback

- “Very interesting course and plenty of resources to look into”
- “Highly knowledgeable trainers”