

2026-03-20

Data Science Training

Nicholas Tierney 

I provide practical and pragmatic courses in R, reproducible workflows, and data visualisation for professionals in government, industry, and research. My mission is to empower people with skills in data science, critical reasoning, and communication, so they can do more.

Course features

- Fewer people, more time with the instructor: Each course has room for only **up to six** people.
- The social aspects of learning: Courses are delivered **live** online or in-person. Learn together with your peers and discuss content together.
- Delivered as interactive practical lessons: I teach, you practice, we discuss.
- Spaced Learning: Courses are delivered as several 1-2 hour lessons over 1-2 weeks, with reading and exercises to complete prior to lessons.
- Ongoing support through office hours. Courses include at least three 60-minute office hour sessions during the 1–2 weeks of lessons and in the weeks that follow. These provide structured time to apply what you’ve learned to your own work, clarify misunderstandings, and get feedback to deepen your understanding.
- Public online teaching materials. Each course has an online book with reading and learning material. For examples of this, see <https://qmd4sci.njtierney.com/> and <https://intro2rpkg.njtierney.com/>. These materials will always be free, and serve as a resource for learners, and also to share this knowledge with the wider community.
- Exclusive content. Participants taking the courses will also gain access to exclusive videos and other materials going into more depth on content.
- Learning scholarships. Every course has one heavily discounted scholarship place (just \$6 AUD) for those who otherwise could not afford to attend. For more details on this, see [pricing](#).

Teaching philosophy

I believe the value in courses comes from the time **you put aside to learn**, and the time **with the instructor**. I would like for people to be able to learn from these materials for free, if they wish. So, I have ensured each course has forever free, publicly available resources. I believe this is a sustainable model as it provides something of value to the community, and also serves to advertise the courses.

Course location and timings

I am based in nipaluna, lutruwita (Hobart, Tasmania, Australia). Courses will be delivered online, although condensed in-person sessions can be arranged in Australia or internationally. Courses are delivered at times to primarily suit those in Australia, although depending on our evening or morning, this may suit you if you are elsewhere in the world. See time differences between [Auckland](#), [Hobart](#), [Brisbane](#), [Perth](#), [New Delhi](#), [London](#), [New York](#), and [Los Angeles](#).

Courses on offer

I currently offer the following courses:

- [R best practices](#)
- [Prepare reproducible and compelling documents with Quarto](#)
- [How to create better graphics using R](#)
- [Level up your R programming by learning functions and debugging](#)
- [Efficiently track changes and collaborate with Git and GitHub](#)
- [Save time and ensure reproducibility using targets and geotargets](#)
- [Learn the introductory fundamentals of R package development](#)

Example schedule

Here is an example delivery timing for the “Quarto for Scientists”:

- Week 1:
 - Getting started and setup (1-2h)
 - Rendering and outputs (1h)
 - Figures and tables (1h)
 - Debugging (how to get unstuck) (1h)
 - End of week 1 office hour and review (1h)
- Week 2:
 - Equations, bibliography, reference systems (1h)
 - Advanced formats and Open discussion (1h)
 - Submit example reports/documents for feedback (1h)
- Week 3:
 - Follow up session with Nick (1h)
- Week 4:
 - Follow up session with Nick (1h)

In order to be most flexible, these sessions will be recorded so participants can watch the material again if they wish.

For current pricing and availability, see the [pricing page](#) or contact me at info@njtierney.com.

Course learning outcomes

Here is a summary of each of the learning outcomes for each of the courses, to see more detail on the courses, click the links to the course website.

R Best Practices

<https://r-best-practices.njtierney.com>

Prerequisites

- Basic R programming experience
- Familiarity with writing R scripts
- Experience working on data analysis projects

Learning outcomes

- How to name things effectively
- Using a style guide
- How to refactor your code
- How to review your code and others'
- How to lay out a project so others know how to run your code
- How to make a reproducible example (reprex)

Introduction to Quarto

<https://qmd4sci.njtierney.com>

Prerequisites

- Basic familiarity with R or Python
- Experience writing scripts or data analysis code
- No prior experience with Quarto or R Markdown required

Learning outcomes

- Write your own Quarto document from scratch
- Best practices for project workflow with Quarto
- Rendering Quarto to HTML, PDF, and Word
- Managing dynamic referencing and creating captions for figures and tables
- Managing bibliographies, reference systems
- Handling common errors in Quarto
- Using Quarto to render slides, websites, and books

This course will also end with an (optional) capstone assessment, where you submit a document you created with Quarto to Nick, and he will review the writing, code, and project with you.

Making Better Graphics

<https://better-vis.njtierney.com>

Prerequisites

- Basic R programming experience
- No familiarity with ggplot2 required

Learning outcomes

- Apply the grammar of graphics

- Use different geoms and aesthetics
- Choose the right graphic for the right data type
- Use facets to explore subsets
- Apply principles of plot hierarchy and proximity
- Understand the fundamentals of tidy data
- Link tidy data with ggplot2
- Polish graphics for publication with labels, themes, colours, and legends
- Save ggplots as high quality images
- Use ggplot2 extensions such as patchwork, marquee, and ggrepel
- Apply techniques to improve clarity (overplotting, faceting, highlighting, colour)
- Identify and avoid common pitfalls
- Critique graphics using principles such as data:ink ratio and hierarchy

Introduction to Functions and Debugging

<https://fun2debug.njtierney.com>

Prerequisites

- Basic R programming experience
- Familiarity with R scripts and basic data manipulation
- Experience running R code and encountering errors

Learning outcomes

- How to write functions to:
 - Manage complexity
 - Explain and express ideas
 - Techniques for developing good functions: outside-in, inside-out
 - Avoid repetition
- How to debug functions and troubleshoot common errors
- Best practices for function documentation

Introduction to Git and GitHub

<https://gentle-git.njtierney.com>

Prerequisites

- Basic computer literacy (creating files and folders)
- No prior experience with Git or version control required
- Helpful but not required: some experience with R, Python, or other programming

Learning outcomes

- Experiment freely without fear of breaking working code
- Collaborate with others without stepping on each other's toes
- Share code publicly or privately with colleagues
- Track down when and where bugs were introduced

- Setting up Git and GitHub
- Push your code onto GitHub
- Understand how to use issues to track ideas or problems
- Use branches to manage features or changes
- Basic collaboration workflows (pull requests, merging, merge conflicts)

Using targets and geotargets

<https://gentle-targets.njtierney.com>

Prerequisites

- Experience writing basic R scripts
- Familiarity with writing functions in R
- Experience with spatial data packages (terra, sf) helpful but not required
- Experience in writing your own data analysis

Learning outcomes

- Understand benefits of using a pipeline approach like {targets}
- How to write functions that work in a pipeline
- How to debug common pipeline issues
- How to use {targets} with {geotargets}

Introduction to R packages

<https://intro2rpkgs.njtierney.com>

Prerequisites

- Comfortable with some R fundamentals (data types, functions, reading data)
- Experience writing basic R scripts
- No prior experience with package development required

Learning outcomes

- Create the basic structure of an R package
- Manage dependencies with `usethis` and `devtools`
- Create documentation with `roxygen2`
- Write and run unit tests with `testthat` to verify package functionality
- Use Git and GitHub to put your R package online
- Understand next steps for advanced package development, including:
 - Automatically run tests with continuous integration via GitHub Actions
 - Make your R package easily installable with the [R Universe](#)
 - Create professional package websites using `pkgdown`

About the instructor

Dr. Nicholas (Nick) Tierney is a [Research Software Engineer](#), and freelance consultant with a PhD in Statistics who specialises in data analytics, R package development, and teaching. He wrote his

first R package in 2015, [neato](#), after being inspired by Dr. Hilary Parker's blog post "[writing an R package from scratch](#)".

His academic work has produced several widely-used packages (see [his software page](#)). During his research fellowship at Monash University with [Professor Dianne Cook](#), he developed tools for exploratory data analysis including [visdat](#), [naniar](#), and [brolgar](#). He then went on to work with [Professor Nick Golding](#) at [The Kids Research Institute Australia](#), working as a research software engineer to translate research methods into R packages such as [conmat](#), a tool used in pandemic modelling. He also maintains Nick Golding's [greta](#) R package for statistical modelling using Google's [tensorflow](#).

Nick actively writes about R related projects at his blog, "[credibly curious](#)". When not coding, Nick enjoys outdoor adventures and hiked the entire Pacific Crest Trail in 2023, documenting his journey at [njt.micro.blog](#).