2026-02-18

Data Science Training

Nicholas Tierney

I provide hands-on 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 and critical reasoning, so that they can do more.

Each course has approximately 6 hours of course content delivered as an interactive practical lecture. There are three "office hour" (60 minute) sessions packaged into each course, which provide an opportunity to ask further questions about the course based on your experience using the materials. These are comprised of one during the course, and two in follow up, after the course is complete.

These follow-up office hours provide structured opportunities to practice skills, review concepts, and get help with applying what they have learnt in their own work.

The courses I currently offer are:

- R Best Practices
- Introduction to Quarto (aka "Quarto for Scientists")
- Making Better Graphics
- Introduction to Functions and Debugging
- Introduction to Git and GitHub
- Using targets and geotargets
- Introduction to R package development

Each course will be delivered over several hands-on, practical sessions, which will be spaced out over 1-2 weeks. The learners will have reading and exercises to complete prior to each lesson. As an example, the quarto for scientists course could be delivered like so:

- Week 1:
  - Getting started and setup
  - Rendering and outputs
  - Figures and tables
  - Debugging (how to get unstuck)
  - End of week 1 office hour and review

- Week 2:
  - Equations, bibliography, reference systems
  - Advanced formats and Open discussion
  - Submit example reports/documents for feedback
- Week 3:
  - Follow up session with Nick
- Week 4:
  - Follow up session with Nick

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

Course numbers are limited and capped at 6 people for optimal learning.

For current pricing and availability, see the pricing page or contact me at `info@njtierney.com`.

Learning materials and philosophy on teaching

Each course will have a free online component, and a private component that only those taking the course can access. Each course will also have one scholarship space available for participants who otherwise could not afford to attend.

Free, forever

Each course has an online book with reading and learning material. For examples of this, see https://qmd4sci.njtierney.com/ and https://intro2rpkgs.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.

Learner only content

There will also be content that is only for those taking the course. This includes the online practical sessions, as well as other concept videos introducing key concepts.

Philosophy on teaching

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 so desire. The free resources accompanying each course will then be publicly available as a reference source. I believe this is a sustainable model as it provides something of value to the community, and also serves to advertise the courses.

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

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)

Course website

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

Introduction to Quarto

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.

Course website

https://qmd4sci.njtierney.com

Making Better Graphics

Prerequisites

- Basic R programming experience
- No familiarity with ggplot2 required
- Experience working with data

Learning outcomes

- How to think about creating good plots
- Good vs bad charts

- Data-ink ratio

- Clear message communication

- Hierarchy of plot design (concepts of proximity)
- Fundamentals of ggplot2:
  - Understanding the grammar of graphics
  - Working with different geoms and aesthetics
  - Common pitfalls and how to avoid them
  - Polishing plots for publication
  - Brief introduction to useful common extensions (e.g., patchwork)

Learning Outcomes

By the end of this course, participants will be able to:

1. Understand the grammar of graphics and how ggplot2 implements it
2. Map variables to aesthetics (x, y, colour, fill, shape, size) appropriately
3. Choose appropriate geometries for different data types (points, lines, bars, histograms)
4. Use faceting strategically to create small multiples with `facet_wrap()` and `facet_grid()`
5. Apply the proximity principle when designing multi-variable plots
6. Recognise and create tidy data suitable for ggplot2
7. Reshape data using `pivot_longer()` and `pivot_wider()`
8. Create effective bar plots with proper sorting and positioning
9. Polish plots with labels, themes, colours, and legends
10. Handle overplotting using jitter, transparency, and summaries
11. Save publication-quality figures with `ggsave()`

Course website

https://better-vis.njtierney.com

Introduction to Functions and Debugging

Prerequisites

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

Learning outcomes

In this course you will learn the following ideas on writing good functions:

- 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

Course website

https://fun2debug.njtierney.com

Introduction to Git and GitHub

Prerequisites

- No prior experience with Git or version control required
- Some experience with R, Python, or other programming language

Learning outcomes

By the end of the course you should be able to:

- Set up Git on a project
- Link your local git project with an online project on 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)
- Work on new ideas and code, without the cocnerns of breaking working code
- Collaborate with others without stepping on each other's toes
- Know how to resolve basic git issues
- Identify when and where bugs are introduced

Course website

https://gentle-git.njtierney.com

Using targets and geotargets

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}

Course website

https://gentle-targets.njtierney.com

Introduction to R packages

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

By the end of this course, you will be able to:

- 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 presenter:

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 github). 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. Specifically, conmat a tool used in pandemic modelling, and he maintained Nick Golding's greta R package for statistical modelling using Google's tensorflow.

Beyond coding, 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.