

Building a statistical computing textbook in the Age of AI

Ronald G. Thomas

2026-04-24



Figure 1: Two identical open notebooks side by side, each with a different outline sketched on its page, a single shared pen resting between them: drafting two structurally distinct textbooks at once.

A textbook's subtitle is a commitment. 'In the Age of AI' commits the author to explaining where the human-LLM division of labor actually falls for each topic.

Introduction

I did not really appreciate how much structural decision-making goes into a textbook until I tried to draft two of them at once. A methods volume ('Statistical Computing in the Age of AI') and a workflow companion ('Biostatistics Practicum') were both overdue, and I wanted them to share enough scaffolding that a student could move between them without reorienting.

The turning point came when I stopped treating 'in the Age of AI' as a subtitle and started treating it as a structural obligation. Three short prose sections of 'Prompts to try' at the end of

each chapter was decoration. What the subtitle required was an explicit front-loaded treatment of what the human statistician contributes, paired with an end-of-chapter verification workflow that exercises those contributions.

We walk through the decisions that produced the two books' current state: chapter template, visual design, bibliography growth, and the specific AI-collaboration pattern. The post is a retrospective on one working session, not a finished tutorial. Readers drafting a domain-specific textbook can probably save themselves some of the back-tracking documented here.

More formally, we document here the convergence of the Knowledge Management and LLM-Augmented Editing concerns of the Workflow Construct described in [A Workflow Construct for the Modern Data Scientist](#). Authoring a textbook is the largest-scale knowledge-management task most academics will undertake; doing so 'in the age of AI' surfaces specifically how the LLM-augmented editing layer interacts with the file-system, document, and bibliographic layers below it. The retrospective documented here is one working session worth of evidence on how those layers compose under sustained use.

Motivations

- Existing statistical computing textbooks are excellent but largely predate widespread LLM use. A curriculum written as if 2019 were still the current year trains students for a workplace that no longer exists.
- Graduate biostatistics students arrive with LLMs already in hand. Pretending otherwise produces a textbook that does not match their reading context.
- I wanted a dual-volume approach: a methods book (numerical algorithms, models, bootstrap, Bayesian) and a practicum (reproducibility, Git, Docker, SAS, CDISC). The two should share scaffolding but remain independently readable.
- A peer-program survey (22 US biostat MS programs) had surfaced specific content gaps (missing data, SAS, survival, Bayesian computation) that I wanted to close without restructuring the books.
- I wanted to document the decision-making as it happened so that the next instructor building out a similar text has a worked example to follow.

Objectives

1. Scaffold two Quarto books (a methods volume and a practicum companion) with compatible chapter structure, shared citation conventions, and visually distinct rendering.
2. Establish a 'statistician's contribution' / 'collaborating with an LLM' pattern that earns the 'in the Age of AI' subtitle.
3. Ground chapter outlines in existing lecture materials, a comprehensive peer-MS-program survey, and Jenny Bryan's STAT 545 philosophical framing.
4. Apply front-matter and code-block improvements (code annotations, first-class callouts, typographic differentiation) as a named 'polish' pass.

The process is documented as it unfolded. Corrections and better approaches are welcome; see the contact section at the end.



Figure 2: A stack of index cards fanned out on a desk, each a different section heading arranged in sequence: structural decisions about chapter ordering.

What is a Quarto book?

A Quarto book is a directory of `.qmd` files (Markdown with executable R code chunks) bound together by a `_quarto.yml` configuration. The `quarto` command-line tool renders that configuration to a multi-format output: HTML, PDF, EPUB, and Word. Think of it as a technical-writing Makefile: chapters, parts, cross-references, and output formats are declared in YAML, and `quarto render` produces a website plus a printable PDF from the same source.

The specific convention I follow is the Posit book-family pattern (R for Data Science, Advanced R, R Packages). Each content chapter carries a three-question diagnostic quiz at the top, collapsible ‘check your understanding’ callouts distributed through the content, exercises and a ‘further reading’ block at the foot, and the quiz answers at the very end.

Prerequisites

This process assumed the following:

- **Operating system:** macOS 15.4, though the setup is portable to Linux and Windows.
- **R 4.4+**, RStudio optional.
- **Quarto 1.5+** (the book format changed non-trivially between 1.4 and 1.5; earlier versions will misbehave).
- **Git** and a GitHub account.

- **An LLM** used as a drafting and verification assistant. The session used Claude Code via terminal; ChatGPT or Gemini would work similarly.
- **Existing lecture materials** (speaker notes .md, slides .qmd, R code .R) for the methods volume; these become the source for chapter content.
- **Time required:** roughly 10 hours for the initial scaffold and framework decisions, excluding content porting.

Installation

No new installation was required beyond the existing Quarto and R setup. The zzcollab framework (an opinionated research-compendium scaffold) created the working directory structure automatically. The only unusual tool was a Python helper script for batch edits across 45 chapter files, written ad-hoc during the session.

```
quarto --version
# 1.9.37
R --version
# R version 4.5.3 (2026-02-28)
```

Configuration: the chapter template

The single most consequential decision was the chapter template. After several iterations, the structure below became the pattern for every content chapter in both books.

```
# Chapter structure (applied to every content chapter)
sections:
- Prerequisites                # Advanced-R-style 3-question quiz
- Learning objectives
- Orientation
- The statistician's contribution # Age-of-AI pillar 1
- Content sections              # with Check-your-understanding callouts
- Collaborating with an LLM on X # Age-of-AI pillar 2
- Exercises
- Further reading
- Practice test                 # only where a course test bank exists
- Prerequisites answers         # placed at the very end
```

Two sections earn the subtitle. ‘The statistician’s contribution’ is front-loaded consciousness raising: the two to five decisions a statistician cannot delegate to an LLM. ‘Collaborating with an LLM on X’ closes each chapter with verification prompts, each following a Prompt / What to watch for / Verification triple.

Example of the statistician’s-contribution section for the bootstrap chapter:

```
## The statistician's contribution {#sec-bootstrap-human}
```

Before handing a dataset to a large language model with the instruction to 'bootstrap the standard error', four decisions require the statistician's judgment:

1. Is the bootstrap appropriate for this statistic at all?
2. What is the dependence structure of the data?
3. Which confidence-interval method matches the statistic's behaviour?
4. Is the bootstrap distribution itself plausible?

Each decision gets a paragraph of explanation, with the failure modes named explicitly (extrema, time-series data, skewed bootstrap distributions) and the chapter that follows providing the vocabulary to make each decision.



Figure 3: A single hardcover book propped half-open on a stand, its pages fanned to show both a printed page and a blank facing page: a textbook still being written.

The four structural decisions

The session's work sorted into four structural decisions, each with downstream consequences across 45 chapters.

Decision 1: book boundary and overlap

The methods volume covers statistical-theory content an MS program teaches in a typical one-quarter course: programming in R, numerical linear algebra, optimization, simulation, bootstrap, linear models, GLM, mixed models, survival, Bayesian. The practicum covers the workflow that surrounds the methods: reproducibility infrastructure, Git, Docker, rrtools, renv, Quarto, tidyverse wrangling, CDISC data standards, SAS, and two full case studies.

The boundary is enforced by discipline in the table of contents. Where a chapter in one volume touches a topic covered in the other, the prose uses a textual pointer (‘see the Missing Data chapter of the companion *Biostatistics Practicum* volume’) rather than a Quarto cross-reference. `@sec-*` anchors do not resolve across distinct books, so the pointer has to stay textual.

Decision 2: Age-of-AI framing

The original chapter template ended with a single ‘With AI assistance’ callout containing three prompts. Reviewing the material as a whole showed that three decorative prompts do not earn the ‘in the Age of AI’ subtitle; they behave as a sidebar rather than as part of the intellectual framework.

The replacement is two structured sections per chapter: a front-loaded ‘The statistician’s contribution’ (what the LLM cannot do on the reader’s behalf) and an end-of-chapter ‘Collaborating with an LLM on X’ (what to verify when the LLM produces output). The preface and introduction were rewritten to name the four-category framework (reliable, unreliable, cannot-due-to-missing-context, cannot-due-to-accountability) that organizes the book’s position.

Decision 3: sourcing chapter content

Each chapter’s source material is distributed across three file types. A `speaker_notes_expanded.md` carries the prose narrative. A `lecture*_slides_*.qmd` carries both inline code blocks and ‘dynamic’ `.fragment .question / .answer` pairs. A `lecture*_R_code.R` supplies supplementary code examples. Early chapter ports missed the inline slide code and the fragment pairs entirely because the porting workflow only read the speaker notes and the R file.

The revised workflow reads all three sources. It integrates inline code from the slides rather than from the stand-alone R file, since the two sometimes disagree, and it converts each slide’s dynamic Q&A fragment into a collapsible ‘Check your understanding’ callout placed after the relevant content section.

Decision 4: visual differentiation

Opening the two books in adjacent browser tabs, they were indistinguishable at a glance. Both used the same primary color, the same body face, the same monospace face, the same SCSS rules. The fix was a deliberate typographic split: the methods volume renders body text in Source Serif 4 (fallbacks Charter, Georgia), the practicum renders body text in Source Sans 3 (fallbacks Inter, Helvetica Neue). Monospace stays JetBrains Mono in both. The SCSS change was ten lines per book.

Complementary configuration changes in `_quarto.yml`:

```
format:
  html:
    code-annotations: hover
    code-copy: hover
    code-overflow: wrap
    code-tools:
      source: repo
      toggle: false
    code-block-bg: true
    code-block-border-left: '#1f4e79'
    highlight-style: arrow
```

Each of these settings is a named Quarto feature. `code-annotations: hover` enables numbered-disc annotations inline in code. `code-tools` puts a ‘view source’ chevron on each page that links to the source `.qmd` in the book’s GitHub repository. `code-block-border-left` replaces a hand-rolled SCSS rule with Quarto-native styling synchronized with dark-mode theming. `highlight-style: arrow` is the accessibility-tuned palette and renders legibly in both HTML and PDF.

Verification

After every structural change, both books were re-rendered and the warnings were checked:

```
cd path/to/methods/textbook
quarto render --to html 2>&1 | grep -iE 'error|warn'

cd path/to/practicum
quarto render --to html 2>&1 | grep -iE 'error|warn'
```

Clean output (empty `grep` result) is the signal to continue. Citeproc warnings (‘citation X not found’) always flagged a bibliography gap and were fixed immediately; bib entries were added incrementally as content ported in.

Daily Workflow

The tools that came up most often:

Command	Action
<code>quarto render --to html</code>	Full book render (HTML only, fast)
<code>quarto render path/to/chapter.qmd</code>	Single chapter render for rapid iteration
<code>quarto preview</code>	Live-reload preview in browser
<code>quarto check knitr</code>	Confirm the R engine is configured
<code>grep -cE '^@' references.bib</code>	Count bibliography entries (growth log)

Command	Action
<code>ls [0-9][0-9]-*.qmd + wc -l</code>	Confirm expected chapter count

The bulk batch-edit pattern (rename a heading across 45 chapters, insert a section block after a given anchor) was handled by short ad-hoc Python scripts using `pathlib` and a content-map dictionary. Shell `sed` worked for single-line substitutions but became unmanageable for multi-line insertions.



Figure 4: The angular concrete tiers of UCSD’s Geisel Library seen from below: a scholarly setting for a textbook still taking shape.

Things to Watch Out For

1. **Quarto’s auto-numbering silently breaks if the introduction chapter is not unnumbered.** `00-intro.qmd` without `.unnumbered` gets counted as Chapter 1, shifting every subsequent chapter’s rendered number up by one. Filename prefix says `09-bootstrap.qmd`, browser says ‘Chapter 10’. Symptom: every cross-reference in the text reads one number off. Fix: `# Introduction {#sec-intro .unnumbered}`.
2. **Subsection numeric prefixes collide with Quarto’s auto-numbering.** Writing `### 1. Correlation and the Fisher-z transformation` renders as ‘9.12.1 1. Correlation...’. Fix: drop the prefix, let Quarto number.
3. **Missing bibliography entries warn but do not fail.** `[WARNING] Citeproc: citation mcelreath2020rethinking not found` is easy to miss in a full-render log. Pipe the render through `grep -iE 'error|warn'` every time.

4. **Cross-book references do not resolve.** @sec-cdisc in the methods volume cannot find the anchor if that anchor lives in the practicum volume. Fix: use prose pointers (‘see the CDISC chapter of the companion volume’).
5. **The Quiz answers heading label mismatches the Prerequisites quiz label.** Advanced R uses ‘Quiz’ / ‘Quiz answers’; my chapters used ‘Prerequisites’ / ‘Quiz answers’, which readers found confusing. Fix: rename the bottom section ‘Prerequisites answers’ and move it to the very end of the chapter (after the Practice test, where one exists).
6. **Speaker notes, slide deck, and R code file each carry different content for the same lecture.** Slide .qmd files are the primary source for inline code; the stand-alone .R file often omits blocks that only appear inline in the deck. Read all three.
7. **Dynamic .fragment .question / .answer blocks in slides are easy to miss.** These are the interactive ‘check your understanding’ moments in a live lecture. In book form they become collapsible callouts. Grep for fragment .question across the lecture directories to find them.

Uninstall / Rollback

Nothing installed that needed rollback. All work is in Git; `git reset --hard <commit>` undoes any session.

What Did We Learn?

Lessons Learned

Conceptual Understanding:

- A subtitle is a structural commitment, not a tagline. ‘In the Age of AI’ obliged the book to explain where the human-LLM division of labor falls for each topic. Three prose prompts at the end of a chapter did not discharge the commitment.
- Two books are easier to draft than one, because the scope limits become explicit. Each volume has a boundary, and content that straddles the boundary gets a textual pointer rather than a cross-book link.
- Peer programs converge on a fairly consistent MS curriculum. A 22-program survey found 17/22 require SAS, 14/22 require missing data, 13/22 require Bayesian computation. Content gaps close faster when the reference distribution is visible.
- The Posit book family (r4ds, r-pkgs, Advanced R, ggplot2) is an excellent structural template, but its conventions do not by themselves earn an ‘Age of AI’ subtitle. That work is additive.

Technical Skills:

- Quarto’s code-annotation feature is under-used. It is the single most distinctive affordance for a computational textbook and was off by default in both books.
- Visual differentiation costs almost nothing: a ten-line SCSS change gives each of two sibling books a distinct body face.

- Collapsible callouts (`.callout-tip collapse='true'`) are the ideal home for slide-style ‘check your understanding’ fragments in a book context. The reader sees the question, thinks, then clicks to reveal the answer.
- Bibliography growth should track content porting. Retrofitting a `references.bib` at the end is painful; adding entries as chapters fill in is trivial.

Gotchas and Pitfalls:

- The `.unnumbered` class on introduction chapters is load-bearing. Without it, every subsequent chapter renders with a shifted number.
- Subsection numeric prefixes duplicate Quarto’s auto-numbering (‘9.12.1 1.’). Let Quarto do the numbering.
- Tidyverse-style `@sec-*` cross-references are invisible in a rendered TOC. They resolve correctly but only in the body text.
- Cross-book `@sec-*` anchors do not resolve. Use prose pointers.

Limitations

- Single author, not peer-reviewed. Content decisions were informed by an LLM assistant but not by a disciplinary editorial board.
- Most chapters are still scaffolds. The skeleton (Prerequisites, Learning Objectives, Exercises, Further Reading, Quiz answers) is complete, but the interior content sections are TODO placeholders in all but a handful of chapters.
- The bootstrap chapter was ported end-to-end as a reference implementation. The remaining 19 methods chapters and most practicum chapters await a similar pass.
- Practice tests are only present in chapters where the course’s test bank contains matching material. New chapters (Bayesian, survival) have no practice test by design.
- No automated progress tracking. A simple markdown table mapping chapter to ‘scaffold / ported / polished’ status would make collaboration easier.

Opportunities for Improvement

1. Port the remaining 19 content chapters’ speaker notes, slide code, and dynamic Q&A fragments into chapter bodies following the bootstrap chapter’s workflow.
2. Implement the four named callouts (`Note` / `Verification` / `Pitfall` / `LLM prompt`) consistently, migrating existing prose callouts as each chapter is touched.
3. Migrate the ‘Check your understanding’ Q&A fragments from untitled `callout-tip collapse='true'` to the named `callout-tip title='Check'` form so they pattern-match across chapters.
4. Add a colophon page to each book documenting the build environment, theme, and font choices.
5. Source hero and portrait images (Efron, Bates, Bryan, Wickham, Marwick) and drop them at `images/cover.png` and `images/portraits/*.jpg`.
6. Push the two book repositories to GitHub (`rgt47/scai`, `rgt47/practicum`) so the `code-tools` ‘view source’ chevrons resolve.
7. Configure Netlify deploys at `rgtlab.org/scai` and `rgtlab.org/practicum` via a path-based proxy from the main site.

Wrapping Up

A textbook's subtitle is a commitment. 'Statistical Computing in the Age of AI' obliged the book to explain where the human-LLM division of labor falls for each topic, and the explanation had to be structural rather than decorative. The current template delivers that through two sections per chapter (statistician's contribution plus collaborating with an LLM) backed by a four-category framework in the introduction.

The secondary lesson was that a dual-volume approach clarifies scope. One book tries to be everything to everyone; two books can afford to be boundary-conscious, each pointing to the other where their topics connect. The methods volume covers what modeling looks like; the practicum covers everything that surrounds modeling. A student who reads both comes out with a complete picture and has an easier time than the single-volume reader because each book's scope is visibly narrower.

In conclusion, four points merit emphasis. First, a subtitle is a structural obligation: it must be earned with sections, not sidebars. Second, two books with shared scaffolding and distinct visual identity are easier to maintain than one oversized volume. Third, Quarto's code-annotation, code-tools, and first-class callout features should be configured on day one, not bolted on during a polish pass. Fourth, when porting lecture content, the slides, the speaker notes, and the R scripts should be read separately, as they do not fully overlap.

See Also

Related posts:

- [Provisioning AWS EC2 Instances](#): the post template this one follows.
- [Blog Post Template](#): the data-analysis sibling template.

Key resources:

- [Quarto Books documentation](#)
- [Advanced R by Hadley Wickham](#): the chapter-structure reference.
- [R for Data Science, 2nd ed.](#): tidyverse applied reference.
- [STAT 545 by Jenny Bryan and Derek Stephens](#): 'everything in data analysis except modeling' framing.
- [zzcollab framework](#): the five-pillar research compendium scaffold used as this post's working directory.

Reproducibility

Tested configuration:

Component	Version
Operating system	macOS 15.4
Quarto	1.9.37
R	4.5.3
knitr	1.51
rmarkdown	2.31
Shell	zsh 5.9
Last verified	2026-04-24

Book repositories:

- Methods volume: `~/prj/tch/methods-textbook/textbook/`
- Practicum volume: `~/prj/tch/biostatistics-practicum/`

To reproduce the two-book scaffold:

```
# Methods volume
mkdir path/to/methods-book && cd path/to/methods-book
quarto create project book
# Edit _quarto.yml with the chapter-template sections.

# Practicum volume (same)
mkdir path/to/practicum-book && cd path/to/practicum-book
quarto create project book
# Edit _quarto.yml with the chapter-template sections.
```

The chapter-template structure (Prerequisites, Learning Objectives, Orientation, The statistician’s contribution, content, Check-your-understanding callouts, Collaborating with an LLM, Exercises, Further reading, Practice test, Prerequisites answers) is a narrative convention, not a Quarto feature; applying it is manual.

Let’s Connect

Have questions, suggestions, or spot an error? Let me know.

- **GitHub:** [rgt47](#)
- **Twitter/X:** [@rgt47](#)
- **LinkedIn:** [Ronald Glenn Thomas](#)
- **Email:** [Contact form](#)

Feedback is welcome in the following cases:

- Drafting a similar domain-specific textbook and wishing to compare structural decisions.
- A better approach to the human-LLM division of labor framework.

- Teaching a biostatistics MS program with different curriculum gaps than the ones this process surfaced.
 - A general connection is of interest.
-

Related posts in this cluster

This post is part of the *Quarto, R Markdown, and Publishing* series. Recommended reading order:

1. [Multi-Language Quarto Documents on macOS](#)
2. [Rapid Conversion of Draft R Scripts to Formal Rmd](#)
3. **[Building a Statistical Computing Textbook](#)** (this post)
4. [Producing R Coding Videos: OBS Screencasts and Narrated Slide Decks](#)
5. [Hosting a Quarto Book Series on Netlify](#)
6. [A Review and Proposal for Policy Regarding Generative AI Use in Graduate Biostatistics Courses](#)