

**Lord, grant me AI workflows,
but not yet**

PHUSE Boston SDE

[About ▾](#)[Members ▾](#)[Grants ▾](#)[Projects ▾](#)[Events ▾](#)[News ▾](#)[Code of
Conduct](#)

Making R Submissions Reviewable for FDA

Practical guidance from R Consortium Submissions Working Group discussion on making R-based regulatory submissions easier to reproduce, inspect, and trust.

[SUBMISSIONS](#)[PHARMA](#)[WORKING GROUPS](#)

AUTHOR

R Consortium Submissions Working Group

PUBLISHED

August 5, 2026

Recently, some sponsors have received FDA feedback asking for SAS-equivalent code for analyses originally submitted in R. For teams investing in open-source tools, that feedback can feel discouraging. But the key message from recent discussion in the R Consortium Submissions Working Group is not that R submissions are infeasible. It is that R submissions need to be easy to reproduce, inspect, and trust.

In an open dialogue with FDA members participating in the working group, reviewers shared that requests for SAS-equivalent code sometimes arise for the following reasons:

- Technical difficulties running the submitted R code
- Differences in statistical methodology between SAS and R

Making R Submissions Reviewable for FDA

Practical guidance from R Consortium Submissions Working Group discussion on making R-based regulatory submissions easier to reproduce, inspect, and trust.

SUBMISSIONS PHARMA WORKING GROUPS

AUTHOR
R Consortium Submissions Working Group

PUBLISHED
August 5, 2026

Recently, some sponsors have received FDA feedback asking for SAS-equivalent code for analyses originally submitted in R. For teams investing in open-source tools, that feedback can feel discouraging. But the key message from recent discussion in the R Consortium Submissions Working Group is not that R submissions are infeasible. It is that R submissions need to be easy to reproduce, inspect, and trust.

In an open dialogue with FDA members participating in the working group, reviewers shared that requests for SAS-equivalent code sometimes arise for the following reasons:

- Technical difficulties running the submitted R code
- Differences in statistical methodology between SAS and R
- Greater reviewer familiarity with SAS

While FDA does not endorse any specific statistical software, SAS has historically been the dominant statistical software for regulatory review, and that context shapes the practical challenges reviewers face. In short, reviewers have sometimes struggled to reliably recreate a sponsor’s environment, install required packages, run scripts without sponsor-specific assumptions, or validate R results.

This post reflects discussion in the [R Consortium Submissions Working Group meeting on June 5, 2026](#) and should not be read as official FDA guidance.

R Is Feasible, But Reproducibility Comes First

R offers transparency, a rich package ecosystem, and the benefits of an open-source language where code, methods, and errors can be inspected directly. As FDA’s [Statistical Software Clarifying Statement \(PDF\)](#) notes, sponsors should consult with FDA review teams early, ensure that software used for data management and statistical analysis is reliable, and have documentation of appropriate software testing procedures available.

Those benefits only translate into a successful submission when the package can actually be run in the review environment. Good practice for R submissions is therefore not simply “use R.” It is “submit R in a way that supports efficient independent review.”



“Recently, some sponsors have received FDA feedback asking for SAS-equivalent code for analyses originally submitted in R.”

Issues noted

- No early communication with the review team
- R environment cannot be reproduced
- Proprietary packages are referenced but not submitted
- Hidden setup assumptions (missing env vars, startup files)
- Hard coded paths (Linux system directories, sponsor network drives)
- Complex nested scripts and nested functions

This is a crisis, but not a cataclysm

- The FDA is not saying “stop using R”, or that the technology is not fit for purpose
- The way through is forward, not backward
- Follow the advice/checklist in the blog post
- **But also, we can make better tools!**
 - “Never let a good crisis go to waste”

“The Pit of Success: in stark contrast to a summit, a peak, or a journey across a desert to find victory through many trials and surprises, we want our customers to simply **fall into winning practices** by using our platform and frameworks. To the extent that we make it easy to get into trouble we fail.”

—Rico Mariani, MS Research MindSwap, Oct 2003

What (some) sponsors are asking us

- “How can we use Connect-deployed MCPs to guarantee that Claude Code follows GxP?”
- “Can you have an option that guarantees Posit Assistant doesn’t install new packages?”
- “How do we implement guardrails so our coding agents can do everything but see row-level data?”
- “When will you offer model fine-tuning and hosting so we can create validated models?”

**“Lord, grant me chastity and continence,
but not yet”**

—St. Augustine, *Confessions VIII*

**“Lord, grant me *AI workflows*,
but not yet”**

—Me, today

Please let's not solve this with AI

- If we are already having trouble with our human-authored submissions, why would we enshrine our current practices in AI workflows?
- I prefer we create a “pit of success” for everyone: humans and AI agents, sponsors and reviewers
- I won't try to outline a solution (yet)—let's talk about mindsets instead

Lessons from a career in software engineering

- **Curiosity:** Onions in the varnish, Chesterton's fence
- **Empathy:** "You are not the user"
- **Restraint:** Innovation tokens, second-system syndrome

Curiosity

Onions in the varnish

Be curious about why things are the way they are. Maybe they should change.

Chesterton's fence

A landscape photograph showing a grassy hill under a blue sky with scattered clouds. A dark, winding fence runs across the middle of the hill. Several trees are visible on the left side of the hill.

Chesterton's fence

Be curious about why things are the way they are. Maybe they should stay the same.

Empathy

You are not the user

Just because something makes sense to you, doesn't mean it will make sense to your user (FDA reviewer)

Issues noted

- No early communication with the review team
- R environment cannot be reproduced
- Proprietary packages are referenced but not submitted
- Hidden setup assumptions (missing env vars, startup files)
- Hard coded paths (Linux system directories, sponsor network drives)
- Complex nested scripts and nested functions

**“Always code as if the guy who ends up
[reading] your code will be a violent
psychopath who knows where you live.”**

—John Woods, comp.lang.c++, 1991

Restraint

Innovation tokens

Every time you embrace a novel technology, you've spent a token. Act like you only have ~~three~~ one. Embrace boring technology.

Second-system syndrome

A technologist's second system is when they are most dangerous. It's when they try to realize all the ideas they thought of while implementing the first one.

Lessons from a career in software engineering

- Be **curious** about why things are the way they are. Doggedly pursue the truth, it is usually worth it.
- Get to know your user and treat them with **empathy**. Keep their experience in the forefront of your mind.
- Adopt new technology and techniques with **restraint**. Cleverness should be treated more as vice than virtue.

Finance

Public
Sector

Insurance

Life
Sciences

Retail

Industrials



Finance

Public
Sector

Insurance

Life
Sciences

Retail

Industrials



Finance

Public
Sector

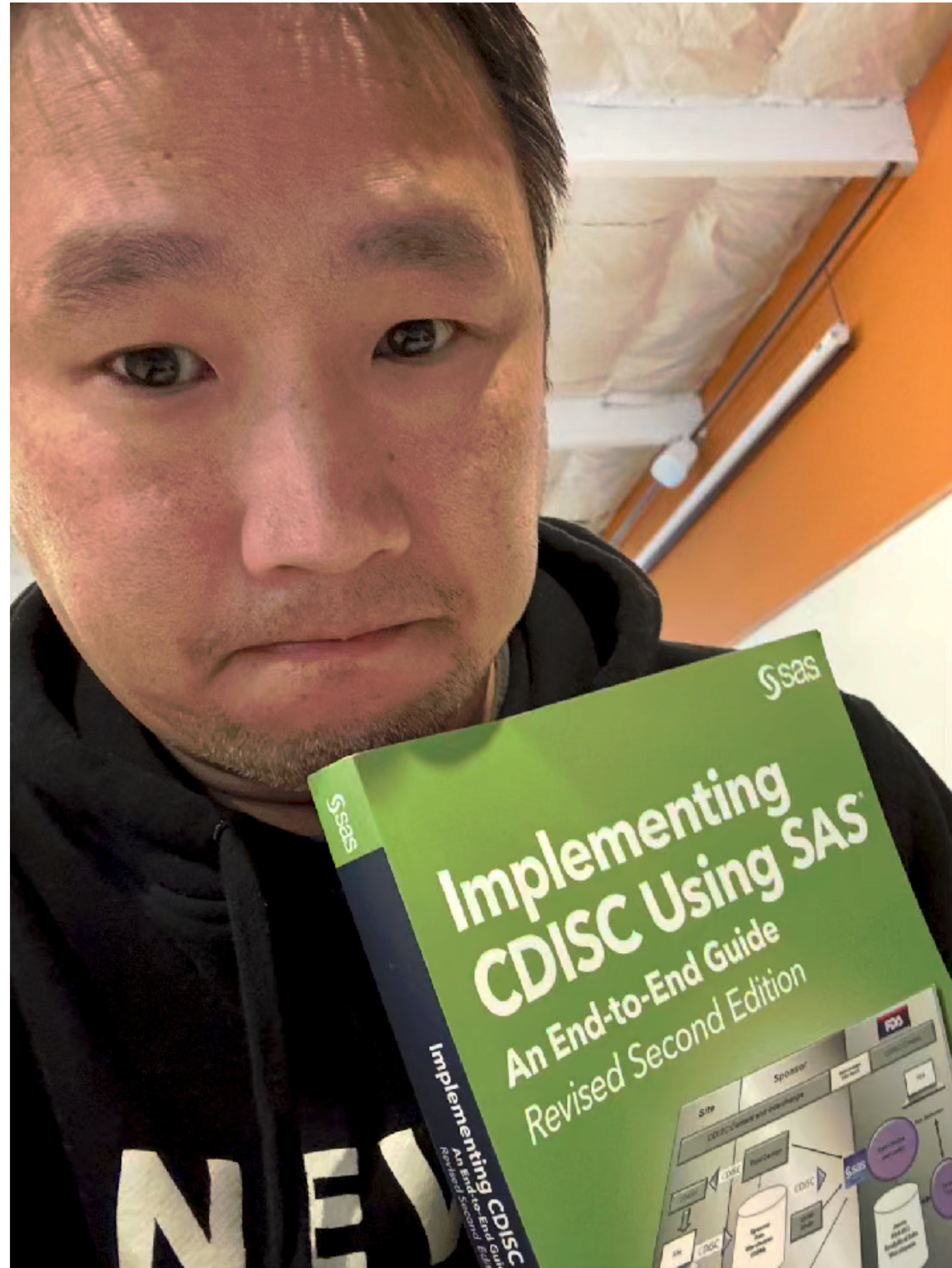
Insurance

Life
Sciences

Retail

Industrials







Adrian Waddell ✓ PhD · 1st

Senior Principal Software Engineer, Pharma & Life Sciences at
Posit PBC

Mississauga, Ontario, Canada · [Contact info](#)

500+ connections



Tareef, Jason and 31 other mutual connections

 Message



University of Waterloo

We are looking for design
partners—reach out!

Preparing for the AI era

- Attending conferences and webinars is not enough; the following steps are essential:
 - In your spare time, **vibe code** something fun, ambitious, and low stakes using a modern coding agent (Claude Code, Codex, Posit Assistant, etc.)
 - Before you think about SKILL.md, MCPs, fine tuning, guardrails...
learn about LLM APIs!
 - It's impossible to form a good mental model if you skip this step!
 - Ask ChatGPT: "Teach me how LLM APIs work. Go step-by-step, with examples."

Be like Eric Ma

ericmjl / llamabot

Type / to search

<> Code

Issues 25

Pull requests 4

Agents

Discussions

Actions

Projects

Security and quality

Insights

llamabot

Public

Sponsor

Watch 3

Fork 35

Star 182

main

25 Branches

236 Tags

Go to file

T

+

<> Code

github-actions and github-actions[bot]

Add release notes for 0.19.16

30d24ae · 2 months ago

2,323 Commits

.agents/skills	feat(mcp): first-class MCP tools + marimo-pair skill	6 months ago
.github	ci: run the test matrix unconditionally on every PR	2 months ago
.llamabot	chore(llamabot): Simplify .gitignore by removing exc...	last year
.vscode	Add MCP server for LlamaBot documentation	last year
archive	feat(experiment_design_agent): Revamp experiment...	11 months ago
docs	Add release notes for 0.19.16	2 months ago
examples	feat(toolbot): Add ToolBot module with documentatio...	last year
llamabot	fix(cli): resolve git paths via GitPython for worktree co...	2 months ago
notebooks	fix: waterfall span visualization accuracy and SimpleBot...	3 months ago
scripts	ci: stabilize import benchmark with warmup + min-of-3 ...	2 months ago
talks	docs(examples): PEP 723 local llamabot; async_simple...	7 months ago

About

Pythonic class-based interface to LLMs

ericmjl.github.io/llamabot/

cli

large-language-models

llm

python

Readme

Activity

182 stars

3 watching

35 forks

Report repository

Releases 171

v0.19.16

Latest

2 months ago

+ 170 releases

Sponsor this project

Thank you!