

DAILY ENGLISH READING

24 August 2026 - 30-minute active reading pack for Davin

TODAY'S QUESTION

Is statistical programming becoming research software engineering?

Difficulty: C1 | SP careers | open source | R/SAS | software engineering

WHY SELECTED

The main peer-reviewed open-access article argues that pharma increasingly needs reusable, validated statistical software rather than repeated study-specific code.

A fresh Novartis role (6 Aug 2026) asks for provenance, orchestration, versioning, open-source statistical tools, cloud reporting, AI and automation.

READING ORDER

0-3 min Preview: Why should pharma invest in statistical software engineering?

3-15 min Main: barriers, validation, reusable packages, collaboration and RSE.

15-20 min Novartis: provenance, orchestration, versioning and AI.

20-25 min CDISC ARS: machine-readable analyses, results and traceability.

25-30 min Output: redesign one repeated SP task as reusable software.

SOURCES

Main: The statistical software revolution in pharmaceutical development

PMC: <https://pmc.ncbi.nlm.nih.gov/articles/PMC12989278/>

Support: Novartis - Associate Director, AQS Tech & Scientific Computing

Novartis career search: req-10085250-associate-director-aqs-tech-scientific-computing

Support: CDISC Analysis Results Standard (ARS)

CDISC: [/standards/foundational/analysis-results-standard](#)

BRIEF BACKGROUND

Open-source statistical development lets methods be implemented once, tested deeply, reused across studies and improved collaboratively. Sustainable packages require licensing, documentation, testing, stable APIs, maintenance and governance.

The paper highlights rbmi, crmPack, bonsaiforest and tern as reusable examples.

Regulatory acceptability depends on documented reliability and testing, not one mandated programming language. Risk-based validation scales evidence to intended use.

A modern SP stack increasingly includes versioning, orchestration and provenance.

KEY VOCABULARY

research software engineering	科研软件工程	reliable reusable software for scientific/statistical methods
open-source license	开源许可证	license permitting defined use modification and redistribution
validated infrastructure	经过验证的基础设施	controlled environment with documented testing and intended use
reusable package	可复用软件包	maintained component solving recurring problems across projects
regulatory submission	监管申报	formal evidence package submitted to a health authority
risk-based validation	基于风险的验证	validation effort scaled to intended use and failure risk
provenance	来源与执行谱系	record of data code versions and steps behind a result
orchestration	流程编排	coordination of programs data assets outputs and dependencies
versioning	版本管理	controlled tracking of code data metadata and output versions
dependency management	依赖管理	control of library versions required for reproducible execution
API design	应用程序接口设计	stable understandable interfaces for software components
cross-company collaboration	跨公司协作	organizations jointly developing shared software infrastructure
software lifecycle	软件生命周期	design development testing release maintenance and retirement
reproducibility by design	设计即复现	building reproducibility into the system from the beginning
hybrid programming stack	混合编程技术栈	controlled use of SAS R Python and other tools together

USEFUL PHRASES

1. move from one-off scripts to reusable software
2. treat statistical software as regulated infrastructure
3. document package versions and build information
4. validate software according to its intended use
5. share the maintenance burden across organizations
6. separate scientific innovation from repetitive implementation
7. orchestrate data, code and outputs as one workflow
8. make provenance available from planning through submission
9. support both legacy and open-source tools
10. build career paths for statistical software developers

COMPREHENSION QUESTIONS

1. Why does pharma need dedicated statistical software engineering?
2. What makes an open-source package sustainable rather than merely public?
3. Why is risk-based validation better than focusing on one language?
4. What do provenance, orchestration and versioning reveal about modern platforms?
5. How can reusable packages make AI-assisted programming safer?

RETELLING PROMPTS

30 sec: One-off programs -> reusable statistical software.

45 sec: Package -> validation -> version -> orchestration -> report -> provenance.

60 sec: Explain how SAS, R, open source and AI can coexist in the future SP role.

5-MINUTE SPEAKING / OUTPUT TASK

Minute 1: Choose formatting, TEAE counts, date imputation, baseline, visit mapping or Listing generation.

Minutes 2-3: Define inputs, outputs, API, parameters, tests and examples.

Minute 4: Add versioning, regression tests, dependencies, validation evidence and rollback.

Minute 5: Explain how AI could use the component without inventing its logic from scratch.

ONE SENTENCE TO KEEP

**The next stage of statistical programming is not simply choosing SAS, R, Python, or AI;
it is building reusable, validated and traceable software systems in which each tool has a controlled role.**