The 10th Data Science Roundtable Discussion, Results of the OSS and R Discussion

English translated version

Disclaimer

The content of this discussion represents the opinions exchanged among participants.
It does not constitute the official position of their respective organizations.

High Level Summary (1)

- Theme
  - Potential and Challenges of Utilizing R in the Pharmaceutical Industry

- Significance and rationale for selecting the theme
  - While interest in open-source software (OSS), particularly R, is growing within the pharmaceutical industry, there remains a lack of well-organized materials and shared understanding regarding the systematic implementation of R under GxP. This theme aims to build a common understanding by progressively discussing key issues toward the effective utilization of R in the pharmaceutical industry, starting with identifying the benefits and risks of R adoption, followed by countermeasures against the identified risks (particularly measures and challenges related to quality), the role of the data science department and the activities required for in-house implementation, and further topics such as ensuring accuracy, reproducibility, and traceability, as well as R/package management.

- Information provided to participants before and on the day of the meeting
  - Food and Drug Administration. “Computer Software Assurance for Production and Quality Management System Software” (2026)
    - https://www.fda.gov/regulatory-information/search-fda-guidance-documents/computer-software-assurance-production-and-quality-management-system-software
  - The reports by JPMA
    - https://www.jpma.or.jp/information/evaluation/results/allotment/DS_202412_oss_questionaire.html
    - https://www.jpma.or.jp/information/evaluation/results/allotment/DS_202406_OSS.html

High Level Summary (2)

- Agenda items and discussion topics on the day
  - First, the participants discussed why the data analyses in clinical development are conducted through programming and leaned the importance of accuracy, reproducibility and traceability for the programming in the clinical development. In addition, the participants discussed why R was being introduced into the clinical development. After that, with these discussion results, the participants learned CSV and CSA, and discussed what quality assurance activities are required for R and R packages.

- Significance and value of the discussions conducted
  - The main participants were the statistical programmers in pharmaceutical companies and CROs, not the specialists of CSV. Thus, the discussion enhanced the participants' understanding the concepts of CSV and CSA. In addition, the participants understood what quality assurance activities were needed for R packages depending on their intended use and context, that is, the purpose and usage of R and R packages and how certain CSV-related activities can be leveraged based on CSA principles. Furthermore, participants recognized the need to proactively deepen their understanding of R and R packages and begin implementing appropriate quality assurance activities within their own organizations while continuing to monitor efforts toward industry-wide standards.

Summary of the One-Day Discussions by Group

Groups 1 to 4

Group 1

1. GCP Operations and Programming

- Regulatory authorities expect programming that can be understood by humans.
- Processes must be reproducible regardless of who executes them.
- GUI-only software is generally not acceptable.
- SAS and R are used for CSR-related analyses.
- WinNonlin is used for PK analyses.

Group 1

2. OSS Available at low cost, flexible, and usable long-term in many cases

	Advantages	Disadvantages
Accuracy	Supported by a broad community Major tools are used worldwide	CSV by the users
Reproducibility	Logic can be retained in code	Version management (backward compatibility challenges)
Traceability	Logic within functions can be traced	Methods for log review need to be established

3. CSV

- Define requirements → validate → document → demonstrate fitness for purpose.
- Validation and requirements of analytical environment.
  - OS and packages: version control.
  - Logs: output from logging packages.
  - Program files: readable without character corruption.
  - Results: consistency with SAS results.

Group 1

4. Process Risk and Leverage

- Consider the impact when software does not behave as intended.
  - High-risk regulatory analyses: scripted.
  - Lower-risk internal analyses: unscripted.

- Leverage
  - Leverage supplements quality assurance.
    - Usage metrics: downloads, books, publications.
    - QC deliverables: CAMIS outputs, GitHub test results.
    - Pharmaverse ecosystem.

Group 2

- Programs must be understandable and interpretable by humans.
  - Stakeholders include self, colleagues, and regulatory authorities.
  - The program and analysis need to support reliability, reproducibility, and accountability.

- CSV for R analytical environment
  - Design the entire R ecosystem.
  - Requirements should be driven by users.

Group 2

- CSV for R Packages
  - Improve efficiency through a CSA-based approach.
  - Open source enables the use of collective intelligence.
  - Community responsibility can be shared and leveraged.

- For high-risk packages, relying only on leverage is dangerous.
  - Organizations should establish their own framework and documentation.
  - A remaining challenge is who approves usage and according to what criteria.

Group 3

Key learnings

- Participants shared similar concerns. The central question was how to approach CSV.
- Defining CSV requirements is difficult and requires experience.

Messages to take away

- Who can decide CSA and leverage? How can CSA leverage be justified without direct review of logic?
- Can regulatory observations from over-reliance on leverage be minimized?
- Could organizations jointly fund validation of R?
- Is it realistic to eliminate SAS completely?

Group 4

- R : Open source and customizable. Requires CSV + QC.
- SAS : Validated and supported. Requires CSV + QC.
- Data analysis
  - Accuracy, reproducibility, and traceability.
- Package-by-package CSV
  - Based on risk.
    - Cosmetic editing versus altering numeric results.
    - Impact on submissions or decision-making.
    - Before or after analysis process.
- Leverage + additional testing.

Topic-Based Discussion Summary

Discussions 1 to 6

Discussion 1

What software used in GCP-related work requires users to write programs or scripts?

Clinical Development and GxP Software

Software (20 cases)

SAS, R, Python, JMP, NONMEM, WinNonlin.
Spotfire, OpenBUGS, JAGS, Stan, Shiny, RAVE.
Pinnacle 21, Excel, VBA, Tableau.
PowerShell, CLI, EDC systems, VTL, Command Prompt.

Major use cases

- Data processing and analysis
  - Data analysis/Interim analysis/Internal and external analysis
  - Analysis and batch processing,
  - PK analysis/Bayesian statistics/Simulation.
- Clinical development/Analysis
  - CSR/CTD TFLs
  - ADaM and SDTM, programming for standard
  - PK analysis, sample size calculations
- Visualization and reports
  - Shiny
  - dashboards, reports.
- Data standard and submission
  - define.xml, Reviewer's Guides
  - Batch file operations.
- Safety, quality and automation
  - Safety monitoring, trial management,
  - QC, validation, automation, utility tool and function

Discussion 2

Advantages and disadvantages of using R for clinical trial analysis from the perspectives of accuracy, reproducibility, and traceability.

Accuracy

Advantages

- Access to cutting-edge statistical methods.
- Source code review is possible.
- Packages and functions can be checked (the correct test data are needed).
- Large user community identifies and fixes issues quickly.
- Code can be modified directly by users.
- Documentation is publicly accessible.

Disadvantages

- Analytical assumptions may not always be transparent.
- Package quality varies.
- No single entity guarantees quality.
- Potential bugs or discrepancies from published methods.
- Re-validation after package updates.
- Additional effort required for validation and correct test data.
- It is difficult to determine the criteria for package selection.

The key advantages are transparency and access to cutting-edge technologies. However, robust governance is required for package selection, validation, updates, and revalidation.

Reproducibility

Advantages

- Environment can be fixed using tools such as renv.
- Git-based version control is available.
- Identical environments produce identical results.
- Version differences are visible.
- Historical environments can often be restored.
- The same codes, packages and versions can be shared and re-executed elsewhere.
- Procedures remain executable as code.

Disadvantages

- Logging and version records must be maintained.
- Both R and package versions must be controlled.
- Different versions may produce different results.
- Frequent updates create maintenance burden.
- OS and dependency consistency is required.
- Support and maintenance for older versions may not always be available.
- Initialization settings can affect results.

Reproducibility requires a controlled environment and comprehensive execution records. The standard operation needs the integration of renv/Git/log.

Traceability

Advantages

- Packages such as logger can provide detailed audit trails.
- Processing steps are retained in code.
- Version controls and execution datetimes and logs become evidence.
- Function-level logic can often be traced.
- Package source code is accessible.
- The retained codes are used to reproduce and trace, like SAS.
- Open source reduces black-box behavior.

Disadvantages

- Environments and dependencies may cause execution issues.
- Traceability depends on governance practices, like Git, naming and log.
- Maintaining logs and evidence requires effort.
- Log reviews can be difficult.
- Hidden assumptions, such as function masking, may reduce transparency and complicate traceability
- Differences between versions need to be appropriately managed.
- Codes become flexible and hard to trace without standardization.

It is not enough to just retain logs for the traceability. The key is to define naming conventions, logging standards, and audit trail retention requirements in advance.

Discussion 3

You are participating as a programmer in a project to build a GxP-compliant R analytical environment.
Identify requirements and verification methods needed to ensure accuracy, reproducibility, and traceability.
Consider how records should be retained.

Requirements for a GxP-Compliant R Environment

Environment and configuration
- Define OS and runtime versions.
- Manage R and package versions.
- Fix and reproduce environments using renv.
- Ensure analysts can access the same environment.

Reproducibility
- Same environments and inputs produce same results.
- Any user can reproduce results.
- Control random seeds.
- Reconcilable to publications or other results like SAS.

Logging and traceability
- Standardized execution logging.
- Change tracking.
- User access control and access logging.
- Protection against unauthorized modification.

Package quality and availability
- Packages are available.
- Ensure the quality of packages.
- Cutting-edge statistical methods are available.
- Packages are validated.

Operation and training
- Rules and instructions are prepared.
- Minimum accesses are given.
- Trainings are done.
- Managed and documented by designated system administrators.”

Accuracy and integrity of deliverables
- R is executed correctly.
- Code returns the expected and correct result.
- Deliverables, programs, data, TLFs, environments are preserved.
- Clarification of encoding settings to prevent character corruption.

Verification Methods

Installation and configuration
- Based on installation instruction, execution testing.
- Installation logs and MD5/checksums etc.
- Documentation of installed package and R versions.
- Recording the packages and versions using renv etc.

Reproducibility testing
- By re-run with identical inputs and environments, same result.
- Verify random seeds.
- Compare with manual calculation, SAS and published references.
- Validation with test data.

Log and audit trail
- Log control and retention.
- Confirm retention of execution and error logs.
- Confirmation on access logs.
- Retention of change-control records.

Code review
- Code review.
- Confirmation on correct code.
- Confirmation by test data.
- Double programming and instruction.

Access, retention and audit trail
- Verify least-privilege access settings.
- Read only snapshot.
- Archival as text file.
- Verify access to files created by other users.

Risk-based operation
- Risk-based test.
- Decision on validation by each package.
- Archival of question, challenge, setting and timing.
- Training records.

Mapping of Requirements to Verification Methods

CSV Perspective	Requirement	Verification Approach
Environment Control & Configuration Management	Manage versions of R/OS/packages, maintain a fixed environment, use internal servers	Freeze the environment using tools such as renv, record package/R/OS versions, retain installation logs
Installation Appropriateness	Required packages can be installed correctly	Execute installation testing according to instructions, verify checksums and installation logs
Reproducibility	Same input and same environment produce identical results regardless of who executes the analysis	Re-run with the same data and code, verify random seeds, confirm consistency of results
Accuracy	R functions correctly and the entire deliverable can be reproduced later	Compare with known results, SAS outputs, manual calculations, and reference publications; use test datasets
Logging / Audit Trail	Execution logs are retained and changes can be traced	Manage and archive logs, review error logs, retain change management records
Access Control / Permissions	Analysis personnel can access the same environment with least-privilege permissions	Review access logs, verify permission settings, confirm accessibility of files created by others when appropriate
Output Archive	Deliverables cannot be modified without authorization	Use read-only snapshots, store outputs in text-based formats, implement Git or other change history mechanisms
Package Quality	Use packages with assured quality and appropriate validation status	Perform validation according to intended use of the package; document issues, settings, and validation activities
Operations / Training	Establish operational rules, instructions, and training	Maintain SOPs/work instructions, perform double programming, keep training records, maintain system administration records
Character Encoding	Prevent character corruption and clearly define encoding standards	Verify OS and encoding settings, validate using test datasets

Discussion 4

Classify R packages into High and Low risk.

Packages evaluated as high risk : Impact to analysis result, CSR and data transformation

As the rationale for the **High** rating, direct impacts on analysis results and submission/reporting deliverables were frequently cited.

Package	Risk	Rationale
survival	High	• Directly affects statistical analysis results. • Essential when conducting survival analyses in R. • Impact is high.
admiral	High	• Improves programming efficiency and simplifies code development.
sdtm.oak	High / Low (evaluations divided)	• Mistakes in derivations could have significant consequences, although the amount of derivation programming is relatively limited.
gtsummary	High	• Outputs contribute directly to tables included in the Clinical Study Report (CSR).

Packages evaluated as low risk : Depends on how to use, output and modify

The main reasons for the **Low** rating were that the impact on the results was not substantive, the issues were correctable, and the outputs were intended for internal materials.

Package	Risk	Rationale
tidyverse	High / Low? (evaluations divided)	• Rarely questioned or challenged. • Commonly used collection of foundational packages. • Errors may affect results and therefore could impact later analyses. • Potential impact exists, but severity is considered relatively low. • Importance may depend on usage (Primary vs. Secondary analyses).
xmpdf	Low	• Can be corrected if issues are found.
labelled	Low	• Used mainly for output generation and labels, therefore considered low impact.
flextable	Low / High / To be discussed	• Used for company-internal document and judgment.
openxlsx	Low	• Does not affect analysis results. • Limited impact on the appearance or presentation of outputs.

Risk Evaluation Criteria (Proposed Approach for Future Alignment)

Evaluation Criterion	Packages	Rationale
Criterion 1: Direct Impact on Analysis Results	survival, gtsummary, sdtm.oak	These packages were identified as higher-risk because they can directly affect analysis results, Clinical Study Report (CSR) outputs, or data transformation outcomes.
Criterion 2: Correctability	xmpdf, openxlsx, labelled	These packages were classified as lower risk because issues can generally be corrected after output generation, and their impact is limited to output appearance, formatting, or labeling rather than essential analytical conclusions.
Criterion 3: Dependency on Scope and Intended Use	tidyverse, flextable	The risk assessment may vary depending on how the package is used, such as for primary versus secondary analyses, or for internal documentation versus decision-making purposes.

Discussion 5

Discuss whether there are quality assurance activities for R packages that can be leveraged in daily work.

Potential Leverage Sources

Category	Sources
1. Adoption and Usage History	• Number of downloads and users • Update frequency and version history • Usage surveys and usage records • Usage within pharmaceutical companies and academia • Adoption and trust within communities such as Pharmaverse
2. Usage in Regulatory Submissions and Operational Practice	• Packages already used in regulatory submissions • Usage in clinical trials and submissions by other organizations • Usage in QC activities (e.g., IP, SCR, BT) • Usage in other products or studies • Internal knowledge base and historical records
3. Publications and External Review	• Usage history and citation counts in scientific publications • Usage in peer-reviewed articles • Examples of usage in conference presentations and research publications for data analysis • Packages referenced in books and journal articles • Independent verification by third parties or user communities
4. Result Comparison and Technical Validity	• Comparison with SAS results • Comparison with analysis tools developed by SAS • Comparative outputs from R vs. WinNonlin and similar tools • Publicly available test results (e.g., GitHub repositories) • Comparison materials and deliverables from initiatives such as CAMIS
5. Utilization of QA/QC and Validation Records	• Quality assurance records generated by other organizations • Reviews using QC checklists • Double programming / independent programming verification • Independent calculation and review by multiple analysts • Installation verification and package functionality testing
6. Developer and Community Credibility	• Developer experience (e.g., number of packages developed) • Adoption by trusted users and large user communities • Evidence and reviews available through CRAN, GitHub, and user communities • Cross-validation with other organizations and packages • Assurance of documentation quality and accuracy

Key Point: The intended direction appears to be to use a combination of actual usage records, literature, result comparisons, QA/QC records, and community reliability for risk-based assurance, rather than relying on a single piece of evidence.

CSA Leverage: Evidence Types

Evidence	Potential Leverage
External / Public Information	• Number of downloads and users • Update frequency and version history • Publications, peer-reviewed articles, and citation counts • Use in regulatory submissions, clinical trials, and other products/studies • Adoption within communities such as Pharmaverse
Technical Comparison and Reproducibility Verification	• Comparison with SAS results • Comparison with existing analytical tools • Comparative outputs from R versus WinNonlin • Publicly available test results (e.g., GitHub repositories) • Verification that outputs are generated as intended
Internal and Third-Party QA/QC	• Quality assurance records and QC checklists • Double programming • Independent calculation review by multiple analysts • Installation and integration verification • Evidence from CRAN, GitHub, and community reviews

Depending on the intended use and associated risk, the above evidence should be combined and used as a basis for determining the extent to which additional testing should be performed.

Discussion 6

Risk assessment and identification on testing to be performed and leverage for R packages.

Package-Specific Testing and Leverage

Package	Testing Activities	Supporting Leverage
tidyverse	QC using actual data; multiple tests using actual case examples and medical records	Refer to publicly available documentation
admiral	Double programming (SAS-to-R and R-to-R comparisons)	Double programming results can be supported by outputs from previous package versions and historical executions
gtsummary	Double programming; testing with a restricted set of packages; testing using dummy datasets	
survival	Comparison against SAS results; confirmation against known-correct datasets; reproduction of published literature results; code review; validation of survival analyses; code verification and documentation of review findings	Review publicly available records; extensive history of use; adoption in scientific literature; ongoing maintenance and bug fixes; association with Base R; download statistics; CRAN review and acceptance
sdtm.oak	Controlled Terminology and Pinnacle 21 checks; reproduction of historical study data; Pinnacle 21 validation of generated datasets; verification of required variables (or core domains); Raw-to-SDTM traceability checks; review of complex transformations; validation of mapping logic; checklist-based review	Use within Pharmaverse projects; Pharmaverse community adoption; use in regulatory submissions by multinational companies; adoption within the CDISC community; double programming; verification that outputs are equivalent to actual SDTM datasets (with consideration of SDTM version compatibility)
flextable	QC review; visual inspection for typographical errors, formatting issues, text strings, and values; comparison between R results and Word outputs; confirmation that outputs match expectations; checklist-based review; reconciliation against output datasets; testing with Word and PowerPoint templates	QC reviews showing no tabular formatting issues; successful operation following Microsoft Office version upgrades; version management practices