

Streamlining Statistical Programming with R:

ADaM and TLF Validation Through a Compliant R Programming Environment

PHUSE APAC Connect 2026

Mangesh Kalsekar

 Bristol Myers Squibb™

Agenda



Background & Motivation



Challenges & Compliance Risks



Compliant R Programming Framework



Practical ADaM Validation Workflow

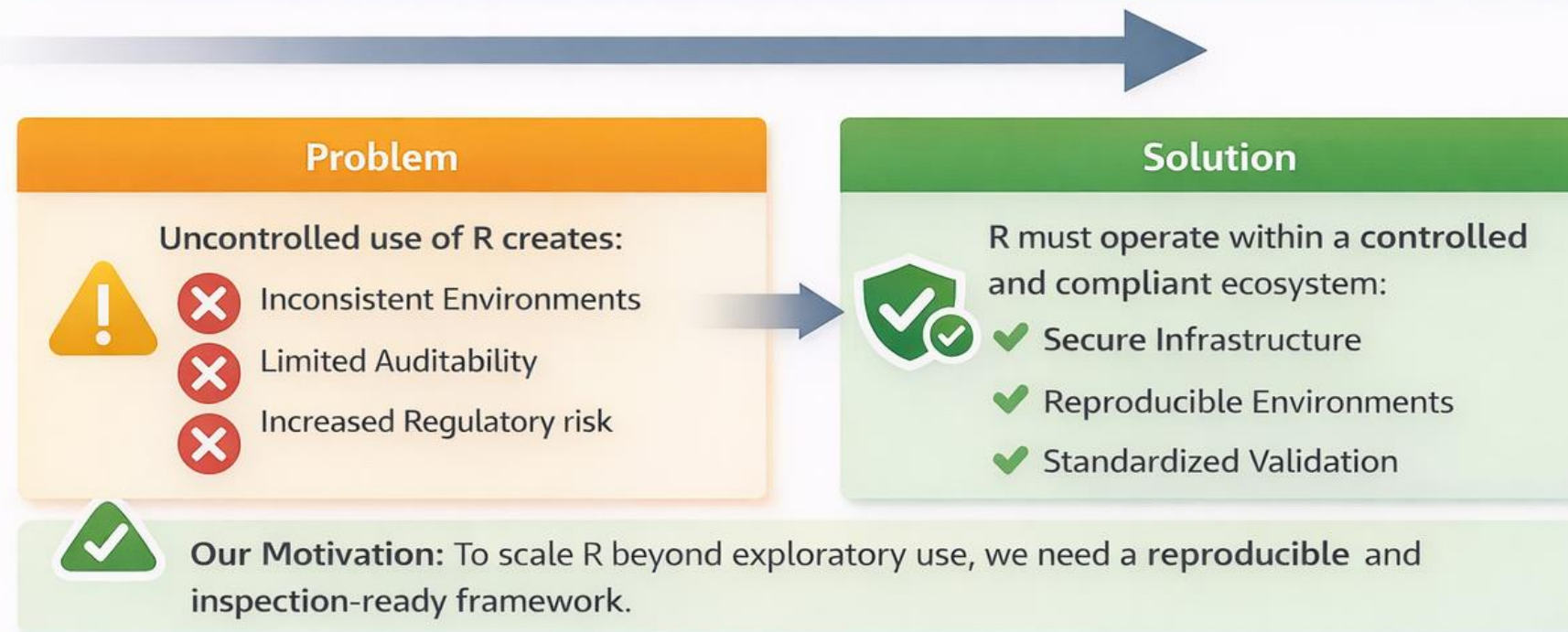


Outcomes & Takeaways

Background

Statistical programming teams must deliver **greater efficiency, transparency, and reproducibility** while meeting **strict regulatory** expectations.

R has **gained momentum** due to its **flexibility, extensibility, and rich open-source ecosystem**.



A controlled R framework is essential to enable **reproducible ADaM and TLF validation** in regulated environments.



Challenges of Using R in Regulated Settings

Key Barriers

- **Inconsistent Package Versioning**

Variability in R and package versions affects reproducibility of historical outputs.

- **Lack of Standardized Logging & Traceability**

Absence of consistent, SAS-like logs limits auditability.

- **Limited Native Validation Capabilities**

No standardized, out-of-the-box ADaM and TLF validation workflows.

- **Open-Source Governance Uncertainty**

Concerns around package approval, lifecycle, and long-term support.



Environment Variability



Missing Audit Trail



Validation Gaps



Regulatory Risk

Impact: Without defined controls, **reproducibility** and **regulatory inspection readiness** are compromised—restricting R primarily to exploratory and non-submission use.

Compliant R Programming Framework



Secure
Infrastructure

Reproducible
Environments

Audit-Ready
Logging

Reusable
Automation

Risk-Based
Governance

Consistent Execution | Transparent Validation | Scalable Adoption

- ❑ **Controlled R ecosystem** with secure infrastructure and enterprise governance
- ❑ **Reproducible environments** using *renv* to lock R and package versions per study
- ❑ **Audit-ready logging** via *logr*, providing SAS-like execution traceability
- ❑ **Reusable automation** enabling standardized, inspection-ready validation workflows

Overview of the R Programming Framework



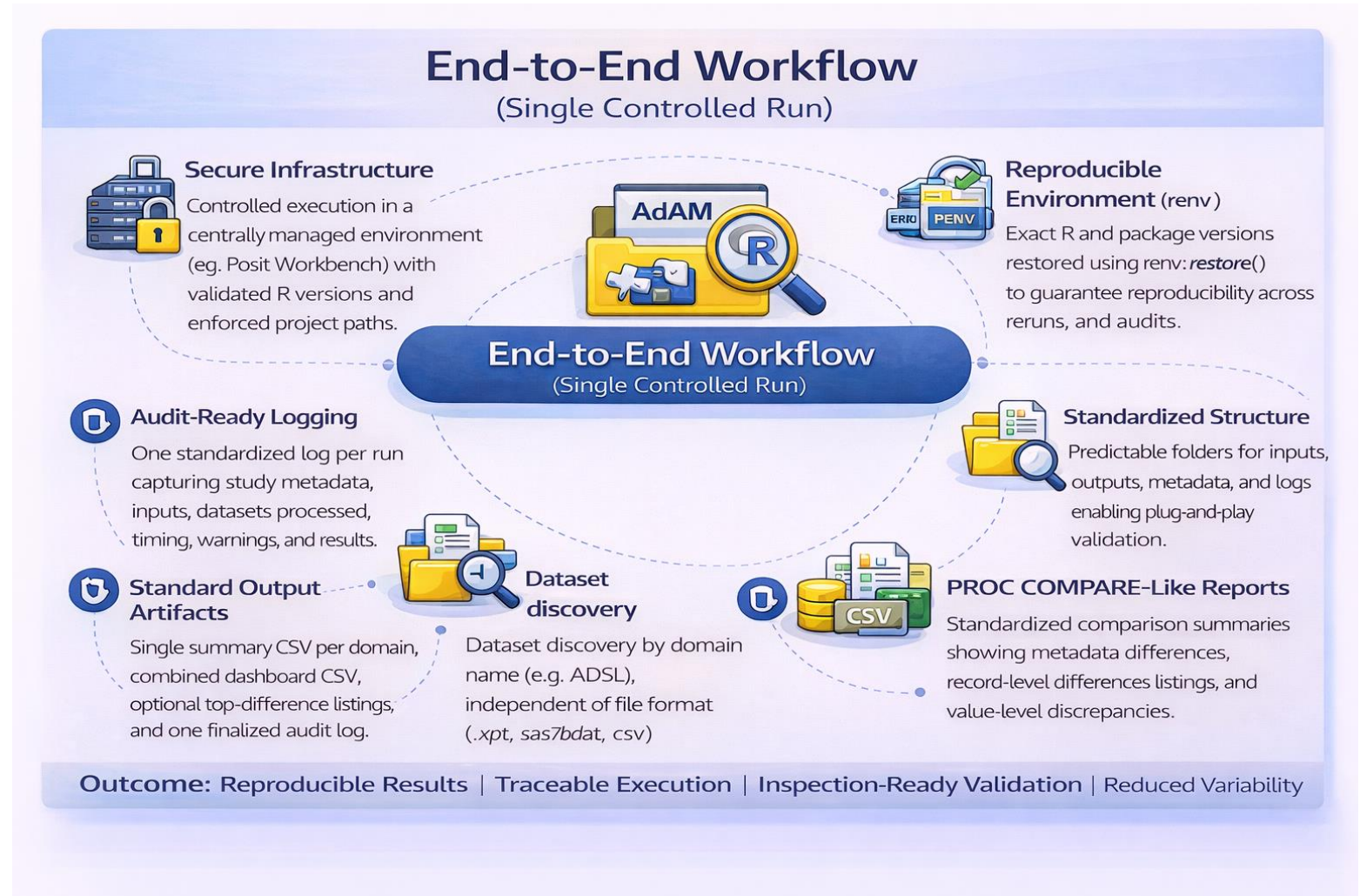
Consistent Results | Traceable Execution | Inspection-Ready Projects | Reduced Variability

- **Standardized Structure**
 - Predefined folders enforce consistent organization of data, outputs, and logs across studies.
- **Controlled Packages**
 - Locked R and package versions ensure reproducible results over time.
- **Audit-Ready Logging**
 - Standardized logs capture execution details to support regulatory traceability.
- **Streamlined Templates**
 - Validated templates reduce manual coding and variability.
- **Streamlined Validation**
 - Built-in workflows support consistent ADaM and TLF validation.
- **Compliance Metadata**
 - Study-level metadata enables governance and inspection readiness.

Practical Technical Example

ADaM/TFL Validation Using the R Programming Framework

-  **Secure Infrastructure**
Controlled execution in a centrally managed environment (e.g., Posit Workbench) with validated R versions and enforced project paths.
-  **Reproducible Environment (renv)**
Exact R and package versions restored using `renv::restore()` to guarantee reproducibility across reruns and audits.
-  **Audit-Ready Logging**
One standardized log per run capturing study metadata, inputs, datasets processed, timing, warnings, and results.
-  **Standardized Structure**
Predictable folders for inputs, outputs, metadata, and logs enabling plug-and-play validation.
-  **Reusable Automation**
Dataset discovery by domain name (e.g., ADSL), independent of file format (.xpt, .sas7bdat, .csv).
-  **PROC COMPARE-Like Reports**
Standardized comparison summaries showing metadata differences, record-level differences by keys, and value-level discrepancies.
-  **Standard Output Artifacts**
Single summary CSV per domain, combined dashboard CSV, optional top-difference listings, and one finalized audit log.



Outcomes & Key Takeaways

Outcomes Achieved



Reproducible Results

Exact R and package versions ensure consistent outputs across reruns and audits.

Traceable Execution



Each validation run produces a complete, standardized audit trail.

Inspection-Ready Validation



Inputs, outputs, logs, and summaries are reviewable and archivable by design.

Reduced Variability



Standardized structure and automation minimize manual and study-to-study differences.

Key Takeaways

- ✓ Governance transforms R from an exploratory tool into a **regulated validation platform**.
- ✓ Standardization enables **scalability** without sacrificing compliance.
- ✓ Validation becomes **repeatable, auditable**, and reviewer-friendly.

- Governance transforms R from an exploratory tool into a **regulated validation platform**.
- Standardization enables **scalability** without sacrificing compliance.

Questions!

Thank You !!!

If you have any questions, please contact

Mangesh Kalsekar Mangesh.Kalsekar@bms.com