Paper OS17

# Harmonizing gsm and the Pharmaverse: Orchestrating SDTM, ADaM, TFLs, and ARS in a Unified R Workflow

Zelos Zhu, Atorus Research, San Francisco, USA
Jeremy Wildfire, Gilead Sciences, Raleigh, USA

## ABSTRACT

Open-source tools for clinical trial reporting have matured within the pharmaverse, offering modular solutions for transforming data into industry standards. In parallel, the gsm suite of packages provides a flexible framework for building pipelines through YAML-driven workflows. We present a gsm–pharmaverse pipeline that integrates *sdtm.oak*, *admiral*, cards, and gtsummary for SDTM, ADaM, ARS, TFLS, respectively, within a connected system of workflows, each supporting its stage. With YAML control files as human-readable checklists, this framework ensures the right tool is applied at the right time, guiding data from raw capture through analysis datasets to final tables.

gsm workflows are both flexible and modular. Workflows can be run by individual contributors (e.g., biostatisticians running scripts) or embedded in automated pipelines (e.g., GitHub Actions or validated Statistical Computing Environments). This approach streamlines reporting, reduces manual effort, and ensures every step is documented and repeatable—strengthening quality oversight and compliance.

## INTRODUCTION

The pharmaverse has rapidly emerged as an open-source standard for clinical trial reporting. Core packages such as sdtm.oak support SDTM transformations, while admiral provides a robust framework for ADaM derivations. Downstream, a rich ecosystem of visualization tools enables production-ready outputs — including tabular summaries with gtsummary and safety visualizations via safetyCharts. At the same time, the CDISC Analysis Results Standard (ARS) is gaining momentum, with packages like cards beginning to support ARS-aligned workflows. Complementing traditional static deliverables, many organizations are increasingly adopting Shiny and web-based review tools to enable interactive exploration of clinical data.

In parallel, the gsm framework — particularly the gsm.core package — has matured into a YAML-driven workflow engine designed to deliver reproducible, traceable, and audit-ready pipelines for clinical data science. The framework currently powers a standardized Risk-Based Quality Monitoring (RBQM) infrastructure, pairing flexible data workflows with robust reporting systems such as those demonstrated in gsm.kri.

Historically, the pharmaverse and gsm ecosystems have evolved along separate trajectories. However, their complementary strengths present a compelling opportunity for integration: combining pharmaverse's domain-specific data standards and reporting capabilities with gsm's workflow orchestration and auditability could form the foundation of a fully traceable, end-to-end clinical reporting pipeline.

With the introduction of {workr} as a standalone workflow engine, we can now explore how these ecosystems can harmonize to deliver cohesive, reproducible pipelines that lower barriers to adopting open-source R infrastructure in regulated clinical trial reporting.

## OBJECTIVE

We present a pipeline that demonstrates how {workr} can orchestrate pharmaverse packages across the full reporting lifecycle — transforming raw eCRF data into SDTM, ADaM, TFL outputs, and ARS-aligned deliverables. Through this case study, we illustrate how harmonizing these ecosystems enables a

cohesive, reproducible workflow that lowers barriers to adopting open-source R infrastructure in regulated clinical trial reporting.

**METHODS**

{workr} engine: In this framework, workflows are defined declaratively using YAML configuration files organized into three sections:

1. Meta — captures descriptive metadata and documentation associated with the workflow
2. Spec — declares required data sources and their structural requirements, defining the inputs needed for downstream derivations and transformations
3. Steps — specifies the execution pipeline. Each step is written as an {output, name, params} block:

```
- output:
  name:
  params:
```

Where, output represents the returned object, name identifies the function being executed, and params contains the function arguments. At runtime, RunWorkflows() parses these step definitions and orchestrates their execution sequentially, effectively reproducing the behavior of a piped R command.

**RESULTS**

Raw → SDTM (workr + sdtm.oak): Raw eCRF data were transformed into SDTM using {workr}-driven workflows aligned with established sdtm.oak implementation patterns. The workflow mirrors the VS domain example from the official vignette, demonstrating how standardized SDTM mappings can be embedded within a reproducible YAML pipeline. The reference implementation is available here.

SDTM → ADaM (workr + admiral): SDTM domains were reintroduced into the workflow engine and processed through admiral derivations to construct ADaM datasets. Functions such as derive_vars_merged() and derive_param_map() generated ADVS parameters and analysis values (e.g., MAP), with derivation logic expressed via !expr tags. This approach preserves transparency while ensuring transformations remain reproducible and audit-ready. The admiral vignette that demonstrates these functions can be found here.

ADaM → TFLs (workr + gtsummary + safetyCharts): The resulting ADaM datasets fed directly into visualization steps, invoking gtsummary for tabular summaries and safetyCharts for graphical outputs. This enables generation of publication-ready TFLs within a fully traceable end-to-end pipeline, adaptable to an organization's visualization standards. Beyond static reporting, the same workflows can automatically refresh web-based or HTML applications as new data are derived, supporting near real-time exploratory analysis and safety review.

ADaM → ARS (workr + cards): In parallel, the derived ADaM datasets were extended using cards to prototype ARS-aligned datasets and outputs. Although the ARS standard is still evolving, this example illustrates how emerging libraries can be integrated into the {workr} framework with minimal friction, enabling early adoption of new reporting standards within an existing, reproducible workflow.

**CONCLUSION**

Each package preserved its modular design, while {workr} provided centralized workflow control, auditability, and reproducibility across the pipeline.

This pharmaverse case study demonstrates the feasibility of a fully open-source, end-to-end clinical reporting workflow. In this model, {workr} serves as the orchestration backbone, while pharmaverse packages deliver domain-specific transformations spanning SDTM, ADaM, TFLs, and ARS. Together,

they form a modular, transparent, and automation-ready reporting architecture. We propose this integration as a blueprint for future collaboration between workflow frameworks and pharmaverse tools, advancing the industry toward interoperable, open-source clinical reporting standards.


## ACKNOWLEDGMENTS

Jon Harmon, Atorus Research
Laura Maxwell, Atorus Research
Natalia Andriychuk, Gilead Sciences
Spencer Childress, Gilead Sciences
admiral development team
sdtm.oak development team
gtsummary development team
cards development team


## CONTACT INFORMATION


Contact the author at:
Author Name: Zelos Zhu
Company: Atorus Research
Phone: 415-823-9682
Email: Zelos.zhu@atorusresearch.com