

BEYOND TLG GENERATION

From reporting specifications to reproducible
and executable clinical reporting workflows

Technical report

Planimeter Kft.

Version 1.0.0

24 September 2026

Methodological demonstration only. The workflow uses Pharmaverse demonstration data and an ADLB-like teaching derivation. Version 1.0.0 also corrects a development-table presentation issue: column N now represents unique subjects rather than longitudinal record counts.

Executive summary

Objective. To demonstrate how a clinical reporting specification can be translated into a reproducible R workflow that connects SDTM-style source data, an analysis-ready laboratory dataset, table shells, executable table layouts, submission-oriented artifacts and review outputs.

Data. The example uses the Pharamverse demonstration datasets `pharmaversesdtm::lb` and `pharmaverseadam::adsl`. Five laboratory parameters are retained: ALT, AST, bilirubin, creatinine and hemoglobin.

Key result. The release contains 9,079 ADLB-like rows from 254 subjects. The ALT shell input contains 665 longitudinal rows and 254 unique subjects. A baseline-to-worst shift dataset contains 1,234 subject-parameter rows, and the abnormality listing contains 30 high post-baseline records.

Interpretation. The technical lesson is not that R can print tables. The lesson is that a reporting specification can become an executable, auditable and reusable asset. This compact package demonstrates only one slice of that architecture; it is not a validated submission system.

Reporting architecture

Layer	Role
1. Source data	SDTM-style laboratory records plus subject-level treatment and population data.
2. Analysis layer	Derive baseline, change, percent change, analysis flags and normal-range categories.
3. Specification	Define rows, columns, statistics, precision and footnotes in a table shell.
4. Execution	Translate the specification into an rtables layout and populate it from analysis data.
5. Artifacts	Create TLG text outputs, an XPT export, listings, figures and a Quarto review layer.

1. Data and derivation framework

1.1 Source datasets

The source datasets are demonstration assets distributed through the Pharmaverse R ecosystem. The release-preparation export reported 59,580 LB rows and 306 ADSL rows before the laboratory selection and derivation steps. The technical package does not redistribute the upstream package datasets; the preparation script loads them directly from their R packages.

1.2 ADLB-like derivation

The workflow retains ALT, AST, BILI, CREAT and HGB, joins subject-level treatment and safety-population variables, derives analysis day, and selects the last non-missing value on or before treatment start as baseline. Change from baseline is AVAL - BASE, and percent change is $100 \times (AVAL - BASE) / BASE$ when baseline is non-zero.

Normal-range status is derived from the lower and upper reference limits. The resulting compact file is deliberately described as ADLB-like: it is sufficient for this demonstration but does not claim full ADaM metadata, controlled terminology, traceability or submission conformance.

Item	Value
Derived laboratory rows	9,079
Unique subjects	254
Laboratory parameters	5
Safety-population rows	9,079
Baseline ALT observations	254
Week 4 ALT change observations	223
Week 8 ALT change observations	188

1.3 Scope of the example

The associated Insight discusses a broader future state in which shells, metadata, reusable code, submission artifacts and interactive review environments form an executable reporting system. The released code demonstrates a compact shell-driven workflow, not a complete metadata platform.

2. From table shell to populated output

2.1 The shell is the specification

The ALT shell is written before the statistical output is populated. It fixes the requested visits, summary statistics, decimal precision and footnotes. In that sense, the shell acts as a human-readable reporting contract: the program should implement it rather than invent the presentation after the analysis is complete.

2.2 Release ALT table

The development output reproduced all cell-level summaries but exposed a QA issue in the column headers. Because the table was stored in long format, `rtables::add_colcounts()` counted longitudinal rows instead of unique subjects. Version 1.0.0 calculates unique-subject counts explicitly and uses the shell-defined treatment order.

Visit	Placebo N=86	Xanomeline Low Dose N=84	Xanomeline High Dose N=84
Baseline	n=86 17.49 (8.26) med 15.00	n=84 17.89 (8.22) med 16.50	n=84 19.07 (9.92) med 16.00
Week 4	n=79 0.80 (8.50) med 0.00	n=72 -0.62 (4.35) med 0.00	n=72 2.25 (7.00) med 1.00
Week 8	n=72 -0.96 (5.33) med -1.00	n=60 0.03 (5.86) med 0.00	n=56 3.34 (17.22) med 1.00

Cells show n, mean (SD), and median. The underlying values are unchanged from the development output. Only the interpretation of column N and the treatment-column order were corrected during release QA.

2.3 Why this matters

A reporting pipeline can be computationally reproducible and still be semantically wrong.

The column-N issue is a small example of why validation must check that a program implements the specification, not merely that it runs without error.

3. Additional reporting outputs

The same derived laboratory data support multiple regulatory-style views. The release reproduces a baseline laboratory summary, a baseline-to-worst-post-baseline shift table, and a compact patient-level abnormality listing. These outputs illustrate how one controlled analysis layer can feed multiple reporting products.

3.1 Baseline laboratory summary

Baseline summaries are produced for all five retained parameters by planned treatment arm using n, mean (SD), median, minimum and maximum. The same reusable summary function is used in the ALT shell and the baseline table, reducing duplicated formatting logic.

3.2 Shift analysis

The shift input contains 1,234 unique subject-parameter rows. For the teaching example, a single post-baseline category is selected with the generic ordering HIGH > LOW > NORMAL. This is intentionally simple and should not be generalized to production safety reporting, where clinical direction and severity rules can differ by parameter.

3.3 Abnormality listing

The listing retains high post-baseline results, calculates the amount above the upper reference limit, sorts descending, and keeps the first 30 rows. This provides a compact traceability layer from aggregate outputs back to patient-level observations.

Subject	Treatment	Lab	Visit	Value	ULN	Above ULN
01-708-1286	Placebo	AST	WEEK 24	168.00	34.00	134.00
01-705-1186	Placebo	BILI	UNSCHEDULED 4.1	124.83	21.00	103.83
01-705-1186	Placebo	BILI	WEEK 4	124.83	21.00	103.83
01-705-1186	Placebo	AST	WEEK 4	135.00	34.00	101.00
01-705-1310	Xanomeline High Dose	ALT	WEEK 8	129.00	32.00	97.00

3.4 XPT and Quarto

The preparation script writes an XPT representation of the compact ADLB-like data with xportr. The reporting script also writes a Quarto source file that reads the generated tables and figures. These artifacts illustrate how the same specification-driven workflow can support both submission-oriented transport and modern review formats.

4. Validation and release QA

The release preparation performed 83 automated consistency checks. All 83 passed. The checks were performed independently from the original R execution and cover dimensions, derivations, cross-file linkage, range classification, table-cell summaries and listing construction.

4.1 Automated checks

Item	Value
Automated checks	83
PASS	83
Failures	0
CHG derivation	Exact agreement with AVAL - BASE
PCHG derivation	Exact within floating-point tolerance
ALT shell linkage	Baseline, Week 4 and Week 8 values matched source rows
Listing linkage	All released rows matched HIGH post-baseline source observations

4.2 Release QA findings

Finding	Release handling
Table 1 column N	Release code replaces these with unique-subject counts 86/84/84. Cell-level summaries are unchanged.
Treatment column order	Release code uses shell-defined order: Placebo, Low Dose, High Dose.
Worst post-baseline category	Retained for reproducibility and explicitly labelled a demonstration simplification; production logic should be parameter-specific.
Baseline timing	Retained for reproducibility; same-day time-of-dose distinctions cannot be made in this compact example.

4.3 Runtime verification

R is not installed in the release-preparation environment, so the cleaned R scripts were not executed here. The supplied reference files were generated successfully by the user-side R workflow. Their numerical contents were independently recalculated and cross-checked in Python. A final local R run is recommended before public release.

5. Interpretation, limitations and reproducibility

5.1 Interpretation

The demonstration supports a specification-centric view of clinical reporting. A shell establishes what must be reported; reusable code then derives the required analysis quantities, populates the output and produces additional artifacts from the same controlled data layer. This is more robust than treating each table as an isolated programming task.

5.2 Limitations

- The source data are demonstration datasets rather than a real regulatory submission package.
- The ADLB-like derivation is intentionally compact and does not claim full ADaM conformance or metadata traceability.
- Baseline is defined from date-level information only; same-day time-of-dose distinctions are not available in this example.
- The shift-table worst-category rule is a generic teaching simplification and should be replaced by parameter-specific clinical logic in production.
- The table shell is human-readable text rather than a complete structured metadata specification.
- The workflow has not undergone a formal validated-system lifecycle or regulatory qualification.

5.3 Reproducibility package

Version 1.0.0 contains two cleaned R programs: one rebuilds the compact derived data from the upstream Pharmaverse package datasets, and one reproduces the reporting layer from the fixed CSV inputs. The package also includes the source-export reference outputs, data dictionary, validation files, release QA notes, GPL license and SHA-256 checksums.

References and software

- R package survival / Pharmaverse demonstration data ecosystem: upstream source of the teaching datasets used by the example.
- rtables and formatters: reusable table-layout and formatting infrastructure used for the TLG outputs.
- xportr: XPT transport-file writing in the demonstration workflow.
- Quarto: reproducible document layer used for the review-report template.
- Associated Planimeter Insight: Beyond TLG Generation: From Reporting Specifications to Executable Clinical Reporting Systems.

License and AI use

The Planimeter R source code is released under the GNU General Public License v3.0 or later. Upstream datasets and third-party packages remain subject to their own terms.

Generative AI tools were used to support drafting, code refinement, documentation and validation. Planimeter Kft. retained responsibility for the analytical design, final code, interpretation and released materials.