

CDISC & Submission QC in R

SDTM → ADaM → TLF: standards, traceability and the double-programming checklist

Rverse Analytics

The submission data flow. Collected data → **SDTM** (tabulation, by domain) → **ADaM** (analysis-ready, traceable) → **TLFs** (tables, listings, figures) + **define.xml** describing every dataset and variable. Each arrow must be reproducible code, not a manual step.

The layers and their R tooling

Layer	Contents	Key R packages
SDTM	one domain per dataset: DM, AE, LB, VS, EX, DS	sdm.oak, haven
ADaM	ADSL (one row per subject), BDS (ADLB, ADVS...), OCCDS (ADAE)	admiral, admiraldev
TLF	shells → tables, listings, figures	rtables, tern, gtsummary, gt
Delivery	XPT v5 transport + define.xml + reviewer's guide	xportr, metacore, metatools

ADaM naming you must not improvise

Variable	Meaning
USUBJID	unique subject id, identical across all datasets
TRT01P / TRT01A	planned / actual treatment, period 1
TRTSDT, TRTEDT	treatment start and end dates (numeric dates)
SAFFL, ITTFL, PPROTFL	population flags, values Y/N
PARAM, PARAMCD, AVAL	BDS parameter, its code, analysis value
ADT, ADY	analysis date, relative day (no day 0)
ABLFL, BASE, CHG	baseline flag, baseline value, change
AESOC, AEDECOD	MedDRA system organ class, preferred term

Traceability rules

- Every ADaM variable states its **source and derivation** in the spec; admiral calls mirror the spec one-to-one.
- Never overwrite an SDTM value in place — derive a **new** variable and keep the original.
- Population **flags are derived once** in ADSL and merged everywhere else; a table never re-invents its own denominator.
- Version-pin the environment: `renv::snapshot()` per delivery, `sessionInfo()` in every run log.

Double-programming QC checklist

```
# QC programmer: independent code from the spec, then compare
qc <- my_independent_adsl(raw) # from spec, not from production code
prod <- haven::read_xpt("adsl.xpt")

diffdf::diffdf(prod, qc, keys = "USUBJID") # full dataset comparison
waldo::compare(table_prod, table_qc) # TLF object comparison
```

- ☐ QC coded **from the specification**, without reading production code
- ☐ Datasets compared on keys, attributes and values (`diffdf`) — zero unexplained diffs
- ☐ Table numbers traced to the ADaM source with a documented filter (population flag + analysis visit)
- ☐ Big-N in headers equals the population flag count from ADSL, not the rows in the display data
- ☐ Rounding convention fixed and shared (R's `round()` is banker's rounding — decide once, document, wrap)
- ☐ Dates: no imputation without a flag (`ADTF`); partial-date rule matches the spec
- ☐ Log shows package versions, seed (if any), and a clean run — warnings triaged, not suppressed
- ☐ Comparison log filed with the delivery

define.xml essentials

Each dataset: label, structure, keys. Each variable: label, type, length, controlled terminology, **origin** (collected / derived / assigned) and, for derived variables, the derivation the reviewer can follow to your code. `metacore` holds the metadata; `xportr` applies it and writes compliant XPT.

Common rejection triggers

XPT variable names > 8 chars or labels > 40 chars · missing controlled terminology mappings · `USUBJID` mismatch across datasets · flags with values other than Y/N/missing · dataset-label vs define.xml drift · un-flagged imputed dates.

Full walkthroughs: rverseanalytics.com/for-cros · ADaM with admiral, TLFs with rtables and the FDA pilot story on the blog.