

R You Ready for SDTM? Let {sdm.oak} Do the Heavy Lifting

Dr. Sangram Parbhane, Syneos Health, Pune, India

Sandeep Mulay, Syneos Health, Pune, India

DISCLAIMER

The views expressed are solely those of the authors, and the content is provided for informational purposes only.

ABSTRACT

{sdm.oak} is an EDC (Electronic Data Capture systems) and Data Standard agnostic solution that enables the pharmaceutical programming community to develop CDISC SDTM datasets in R. SDTM uses raw datasets as input. These raw datasets can vary greatly depending on the EDC (Electronic Data Capture) system and how the eCRF is designed.

{sdm.oak} is designed to be highly versatile, accommodating varying raw data structures from different EDC systems and external vendors. Moreover, {sdm.oak} is data standards agnostic, meaning it supports both CDISC-defined data collection standards (CDASH) and various proprietary data collection standards defined by pharmaceutical companies. The reusable algorithms concept in {sdm.oak} provides a framework for modular programming and can potentially automate SDTM creation based on the standard SDTM spec. The {sdm.oak} package is now on CRAN, supported by CDISC COSA as part of the CDISC 360 initiative.

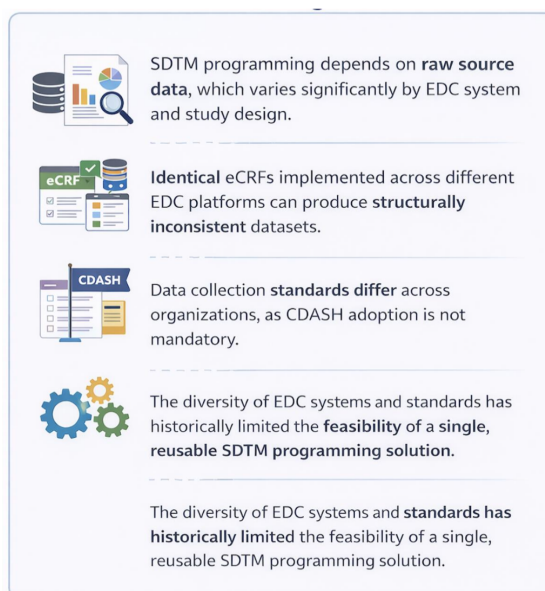
INTRODUCTION

{sdm.oak} v0.2.0, released on 2025-05-22, is now available on CRAN. In this paper, we will introduce the package and showcase its features and present a roadmap for future releases. {sdm.oak} is developed in collaboration with volunteers from several companies, including Roche, Pfizer, GSK, Pattern Institute, Transition Technologies Science, and Atorus Research. {sdm.oak} is also sponsored by CDISC COSA with a vision of being part of CDISC 360 to address end-to-end standards development and implementation.

IN THIS RELEASE

The v0.2.0 release of {sdm.oak} introduces expanded functionality, enabling efficient generation of the DM domain along with additional SDTM domains across the Findings, Events, Findings About, and Intervention classes. Domains and variables still out of scope include Trial Design domains, SV (Subject Visits), SE (Subject Elements), RELREC (Related Records), Associated Person domains, and the EPOCH variable across all domains.

CHALLENGES



CORE CONCEPT – REUSABLE ALGORITHMS

In {sdm.oak}, SDTM mappings that convert collected source data (eDT: External Data Transfer) into the SDTM target data model are organized as reusable algorithms. These algorithms form the foundation of the {sdm.oak} framework.

Key Points:

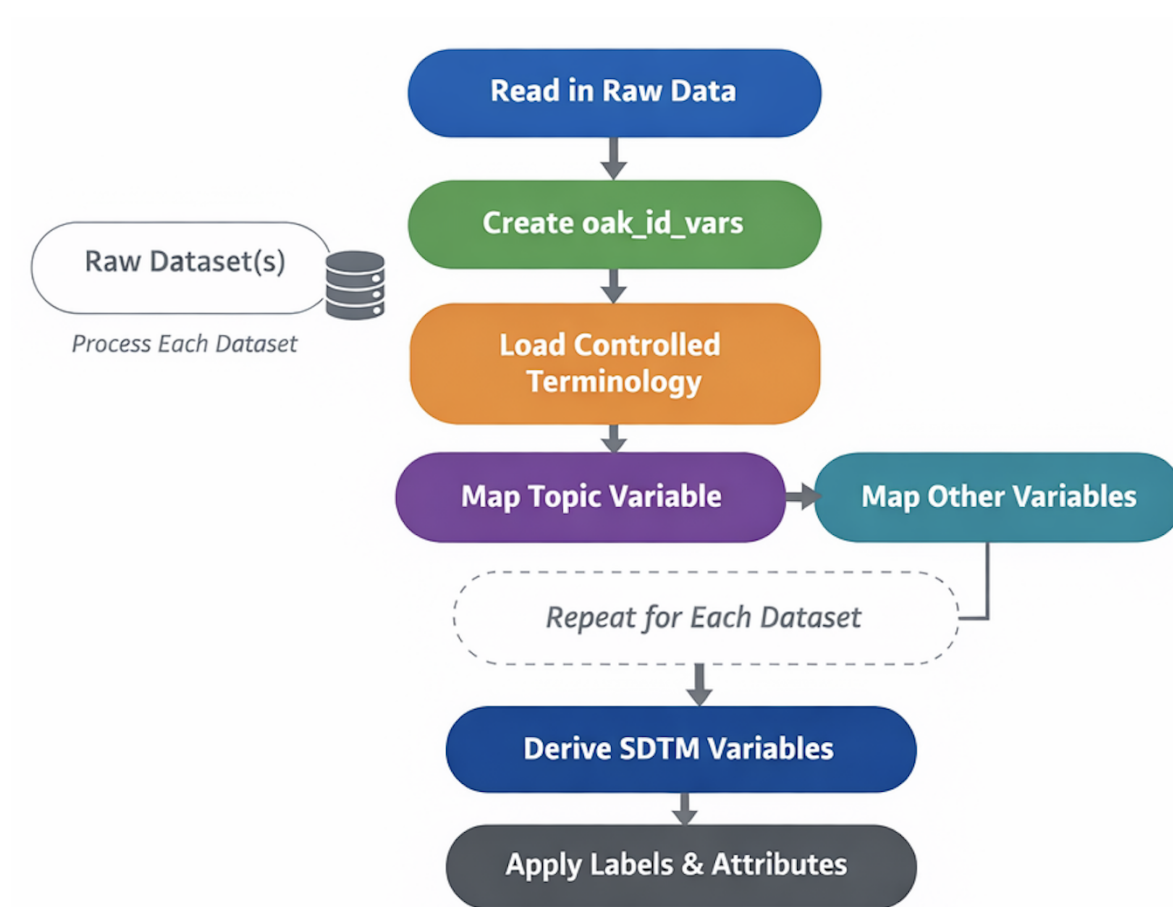
- Algorithms are designed for reuse across multiple SDTM domains.
- The approach is programming-language agnostic and not dependent on a specific implementation language.
- Algorithms are automation-ready and can be predefined for data collection standards within a metadata repository (MDR).

The {sdm.oak} package provides R functions that implement these algorithms. Selected core algorithms are described below to illustrate their reuse across multiple SDTM domains.

Algorithm Name	Description	Example
assign_no_ct	One-to-one mapping between the raw source and a target SDTM variable that has no controlled terminology restrictions. Just a simple assignment statement.	MH.MHTERM AE.AETERM
assign_ct	One-to-one mapping between the raw source and a target SDTM variable that is subject to controlled terminology restrictions. A simple assign statement and applying controlled terminology. This will be used only if the SDTM variable has an associated controlled terminology.	VS.VSPOS VS.VSLAT
assign_datetime	One-to-one mapping between the raw source and a target that involves mapping a Date or time or datetime component. This mapping algorithm also takes care of handling unknown dates and converting them into ISO8601 format.	MH.MHSTDTC AE.AEENDTC
hardcode_ct	Mapping a hardcoded value to a target SDTM variable that is subject to terminology restrictions. This will be used only if the SDTM variable has an associated controlled terminology.	MH.MHPRESP = 'Y' VS.VSTEST = 'Systolic Blood Pressure' VS.VSORRESU = 'mmHg'
hardcode_no_ct	Mapping a hardcoded value to a target SDTM variable that has no terminology restrictions.	FA.FASCAT = 'COVID-19 PROBABLE CASE' CM.CMTRT = 'FLUIDS'

condition_add	Algorithm that is used to filter the source data and/or target domain based on a condition. The mapping will be applied only if the condition is met. This algorithm has to be used in conjunction with other algorithms, that is if the condition is met, perform the mapping using algorithms like assign_ct, assign_no_ct, hardcode_ct, hardcode_no_ct, assign_datetime.	If MDPRIOR == 1 then CM.CMSTRTPT = 'BEFORE'. VS.VSMETHOD when VSTESTCD = 'TEMP' If collected value in raw variable DOS is numeric then CM.CMDOSE If collected value in raw variable MOD is different to CMTRT then map to CM.CMMODIFY
---------------	--	---

PROGRAMMING WORKFLOW



```

library(sdtm.oak)
library(dplyr)

cm_raw <- tibble::tribble(
  ~oak_id, ~raw_source, ~patient_number, ~MDRAW, ~DOSU, ~MDPRIOR,
    1L, "cm_raw", 375L, "BABY ASPIRIN", "mg", 1L,
    2L, "cm_raw", 375L, "CORTISPORIN", "Gram", 0L,
    3L, "cm_raw", 376L, "ASPIRIN", NA, 0L
)

study_ct <- tibble::tribble(
  ~codelist_code, ~term_code, ~term_value, ~collected_value, ~term_preferred_term,
  ~term_synonyms,
    "C71620", "C25613", "%", "%", "Percentage", "Percentage",
    "C71620", "C28253", "mg", "mg", "Milligram", "Milligram",
    "C71620", "C48155", "g", "g", "Gram", "Gram"
)

cm <-
  # Derive topic variable
  # SDTM Mapping - Map the collected value to CM. CMTRT
  assign_no_ct(
    raw_dat = cm_raw,
    raw_var = "MDRAW",
    tgt_var = "CMTRT"
  ) %>%
  # Derive qualifier CMDOSU
  # SDTM Mapping - Map the collected value to CM. CMDOSU
  assign_ct(
    raw_dat = cm_raw,
    raw_var = "DOSU",
    tgt_var = "CMDOSU",

    ct_spec = study_ct,
    ct_clst = "C71620",
    id_vars =
      oak_id_vars()
  ) %>%
  # Derive qualifier CMSTTPT
  # SDTM mapping - If MDPRIOR == 1 then CM.CMSTTPT = 'SCREENING'
  hardcode_no_ct(
    raw_dat = condition_add(cm_raw, MDPRIOR ==
      "1"), raw_var = "MDPRIOR",
    tgt_var = "CMSTTPT",
    tgt_val =
      "SCREENING", id_vars
    = oak_id_vars()
  )

```

Raw datasets and variable names are passed as function arguments. When available in the global environment, the algorithm is executed to derive the target SDTM variable. {sdtm.oak} supports diverse raw input formats and is independent of specific data collection standards, making it both EDC- and standards-agnostic.

ROADMAP

We are planning to develop the below features in the subsequent releases:

- Metadata driven code generation based on the standardized SDTM specification.
- Functions required to program the Domains SV (Subject Visits), SE (Subject Elements) and the EPOCH Variable.
- Functions to derive standard units and results based on metadata.
- Additional features to be developed based on the user feedback.

CONCLUSION

The {sdm.oak} package provides a structured, metadata-driven framework that supports and accelerates SDTM development in R. By standardizing common programming patterns across SDTM domain classes, it enhances consistency, transparency, and efficiency in SDTM dataset creation. Programmer oversight and validation remain essential to ensure data quality and regulatory compliance.

REFERENCES

Conference paper - {sdm.oak} – SDTM programming in R
https://www.lexjansen.com/phuse-us/2025/pp/PAP_PP31.pdf

Ganapathy, Rammprasad. 2024. "Introducing Sdm.oak." October 24, 2024.
https://pharmaverse.github.io/blog/posts/2024-10-24_introducing.../introducing_sdm.oak.html
<https://www.cdisc.org/oak>

ACKNOWLEDGMENTS

We thank Rammprasad Ganapathy, Edgar Manukyan, and Shiyu Chen, along with other contributors and authors, for their contributions to the package.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Dr. Sangram Parbhane
Syneos Health
+91 8605671980
sangram.parbhane@syneoshealth.com
<https://www.syneoshealth.com/>

Sandeep Mulay
Syneos Health
+91 9096115948
sandeep.mulay@syneoshealth.com
<https://www.syneoshealth.com/>