PharmaSUG China 2024 - Paper CC-10017

Practice in Creating an R package for SDTM

En Wang, Longfei Li, Xinyan Lai, Qian Duan, Sanofi China

ABSTRACT

R has achieved significant maturity in the pharmaceutical industry and established well-established ecosystem led by the pharmaverse. SDTM (Study Data Tabulation Model) serves as a crucial component in clinical trial data processing. Converting an existing set of SDTM SAS programs within a company into R programs and further building an R package that includes reusable R functions and documentation, offers an efficient and convenient means of sharing with others. While CDISC is promoting the development of oak on standard CDASH to provide solutions for output SDTM data sets, it's worth considering how to develop internally high-quality R packages or tools to aid in sorting working processes and addressing unique issues for different companies. This paper shares some practices to streamline the development process and enhance the quality of R package.

INTRODUCTION

When transitioning from SAS to R to perform the clinical programming, we faced several challenges in developing an R package for SDTM creation, including:

- Variations in raw data: Different companies have unique raw data sets, which can complicate data cleaning.
- Integrating with existing workflows: We needed to accommodate various established workflows within companies.
- Specialized rules: Issues like re-screening, cut-off data, and various other considerations require specific attention.
- Good Programming Practices: We need to explore and establish a comprehensive set of R's GPP encompassing programming designs, documentation using roxygen2 comments, naming conventions, coding formats, etc.
- Efficient Project Management: Figuring out how to use Git, GitHub, and renv for efficient development management.
- Scientific Validation: Ensuring the performance of R functions through rigorous validation procedures.

GENERAL SDTM FUNCTIONS

SDTM data sets are standardized from raw data, and all the R functions are developed based on tidyverse and pharmaverse, such as stringr, dplyr, tidyr, haven, lubridate, hms, and xportr.

The below lists some common functions for SDTM mapping:

- Read sas7bdata and xpt and write SDTM data sets to xpt
- SDTM metadata specification management
- Parse and convert dates, ISO8601 format
- Sequence number, --SEQ
- Study day, --DY
- Baseline flag, --BLFL
- EPOCH
- Treatment emergent Flag, --TRTEM

- Get the maximum date per participant for RFPENDTC
- Split signal string limited to 200 characters
- Merge supplementary domains to parent data sets

SPECIAL REQUIREMENTS FOR INTERNAL USAGE

Common issues often have rich resources and solutions available, but each project has its own unique set of requirements. It is crucial that we focus our attention on capturing and addressing these special requirements during the development process, as they often determine the overall success of the project.

- The special workflow, particularly notable in Sanofi, involves dividing the SDTM output process into two stages: from raw data to hybrid SDTM and from hybrid SDTM to the final SDTM. In the initial stage, variables and information collected on CRF are directly and promptly exported. Subsequently, based on the hybrid SDTM data sets, more complex variables or observations are derived in the second stage. Notably, hybrid SDTM represents simplified standard information output, while the second stage necessitates deeper consideration of clinical trial design.
- During the development process, we encounter various special issues, such as the calculation of creatinine clearance rate in Laboratory Test Results or LB, the re-screening, the cut-off data, etc. Taking the re-screening problem as an example, in Sanofi, we need to consider three scenarios: a. Multiple screening failures; b. Multiple screenings failure then enrollment; c. Multiple enrollments. To address these scenarios, we not only need to define an additional domain but also need to consider the impact of these re-screening observations on different domains.

Streamlining the process and resolving these unique issues are the focal points of our efforts. In the SDTM R package that we have developed, more than half of the functions are tailored to address these special requirements.

GOOD PROGRAMMING PRACTICE FOR CONSISTENCY AND QUALITY

When it comes to programming in R, it is essential to follow a set of GPP to ensure that your code is efficient, readable, and maintainable. After referring to the tidyverse style guide, Tidy Modeling with R Book Club, and other development experiences such as admiral, we attempted to delve into a set of R's GPP:

- Programming Designs: Although R provides a variety of excellent programming design patterns such as functional programming, object-oriented programming, etc., we have preferred metaprogramming to follow the syntax and usage conventions of dplyr and other tidyverse packages. For instance, the multiple data frame columns as one function argument:

```
# base quoted columns name, sdtm.oak package
source_vars = c("TRTSDTM","ASTDTM","AENDT")
# admiral package
source_vars = exprs(TRTSDTM, ASTDTM, AENDT)
# tidyverse
source_vars = c(TRTSDTM, ASTDTM, AENDT)
```

- Roxygen2 Comments: Not only did we agree on the order of tags, with @title -> @description -> @details ->..., but we also provided detailed requirements for each tag. Taking @title as an example, it should not exceed 12 words, start with a verb, and be capitalized, etc.
- Naming Conventions: Maintaining consistency in naming conventions for functions, arguments, intermediate objects inside functions, and other elements is essential for code readability and aesthetics. For example, start with verbs for function names, use nouns for argument names, and use the same prefix for intermediate objects, etc.

- Profiling and Benchmarking: Using benchmarking to improve the performance for same functionality. For example, obtaining the minimal value across columns. In SAS we can use the `min(V1, V2, V3)` in the data step, there are many ways in R and we will chose to use the concise and efficient method:

```
library(dplyr)
library(purrr)
set.seed(123)
df <- data.frame(
  V1 = runif(3, 1, 10), V2 = runif(3, 1, 10),
  V3 = runif(3, 1, 10), V4 = LETTERS[1:3]
)
bench::mark(
  cbind(df, pmin = do.call(pmin, Filter(is.numeric, df))) |> as_tibble(),
  df |> rowwise() |> mutate(pmin = min(c_across(where(is.numeric)))) |>
ungroup(),
  df |> mutate(pmin = do.call(pmin, keep(df, is.numeric))) |> as_tibble(),
  df |> mutate(pmin = exec(pmin, !!!keep(df, is.numeric))) |> as_tibble(),
  df |> dplyr::mutate(pmin = pmin(!!!rlang::syms(names(purrr::keep(df,
is.numeric))))) |> as_tibble()
)
```

```
#> # A tibble: 5 × 6
#>   expression                              min   median `itr/sec` mem_alloc
`gc/sec`
#>   <bch:expr>                         <bch:tm> <bch:tm>     <dbl> <bch:byt>
<dbl>
#> 1 as_tibble(cbind(df, pmin = do.…  121.7µs  162.4µs     5275.  323.96KB
18.9
#> 2 ungroup(mutate(rowwise(df), pm…   2.65ms   3.23ms      291.     2.6MB
15.2
#> 3 as_tibble(mutate(df, pmin = do…  688.6µs  832.4µs     1103.  895.01KB
14.6
#> 4 as_tibble(mutate(df, pmin = ex…  711.5µs    850µs     1019.   17.75KB
12.5
#> 5 as_tibble(dplyr::mutate(df, pm…  687.7µs  829.1µs     1101.    1.61KB
14.6
```

In addition to the GPP mentioned above, we have also established unified standards for coding formats, comments, unit tests, debugging, controlling argument input, and so on.

PROJECT MANAGEMENT

To achieve high-quality and low-cost programming, we adopted the following excellent tools to manage package development:

- Git and GitHub: Git served as the distributed version control system and was complemented by GitHub to facilitate collaboration. GitHub Issues was fully utilized for issue reporting, tracking, discussing, etc. GitHub Actions was leveraged to automate and streamline the development process through R CMD check, spelling check, coding style, linting, GitHub Pages deployment, etc. However, it's worth noting that GitHub offers numerous other features beyond these two, such as code reviews, wiki pages, project management tools and GitHub Pages for website hosting.
- renv: employing the renv package management system to guarantee reproducibility and preserve consistent environments, while efficiently managing package dependencies.

AUTOMATION FOR SDTM

It can be clearly observed from the special workflow described above that implementing hybrid SDTM automation based on standardized CRF is straightforward. This automated process consists of two integral components: the standard template codes for each domain, which comprehensively cover CRF and rules across various therapeutic areas, and the fact that each individual study has its own dedicated CRF.

Each standard template was developed according to the company-level standard CRF, following the same programming workflow that involves programming headers, loading packages, reading raw data, creating SDTM data shells, implementing transformation rules, and creating final output files. The R comments within the template codes are utilized to annotate CRF pages and source variables, which are subsequently utilized to compare with the study-level CRF to generate the necessary code.

PROGRAMMING DIFFERENCES

As long-time users of SAS, we need to familiarize ourselves with the differences in R programming and gain a deeper understanding of its characteristics, leveraging its strengths such as vectorized nature, functional programming, and the pipe operator. Additionally, there are differences in handling missing values, sorting missing values, and rounding floating numbers in R compared to SAS that we should be aware of, etc. We should embrace these differences and make the most of R's advantages to enhance our data analysis capabilities.

VALIDATION

The bold innovation and meticulous validation are paramount in statistical programming within the pharmaceutical industry. Beyond the commonly utilized testing tools or methods during development, such as R CMD CHECK and unit tests, we also considered two other aspects.

- SAS remains the benchmark for verifying accuracy. For the same study, we utilized both SAS and R to generate SDTM outputs and then compared the results.
- We have utilized the method above to validate R functions in numerous political studies, including oncology phase II, immunology & Inflammation phase III, diabetes phase III, and rare disease phase I/II. In the future, our aim is to further test the accuracy, stability, and efficiency of these functions on increasingly complex clinical trial designs.

CONCLUSION

These practices are part of software engineering, and the new R developers who are used to only be SAS user should have a software engineering mindset and implement the software development techniques to maintain R projects effectively, improve collaboration, reduce errors and increase quality, and ensure robustness and reliability.

REFERENCES

SDTM related packages:

- sdtm.oak https://github.com/pharmaverse/sdtm.oak
- Rsdtm https://github.com/billdenney/Rsdtm
- sdtmval https://github.com/skgithub14/sdtmval
- pharmaversesdtm https://github.com/pharmaverse/pharmaversesdtm
- datacutr https://github.com/pharmaverse/datacutr
- sdtmcheck https://github.com/billdenney/Rsdtm

Good Software Engineering Practice for R Packages. Accessed on July 3,2024. https://github.com/ynsec37/workshop-r-swe

Programming Strategy - admiraldev. Accessed on July 3,2024. https://pharmaverse.github.io/admiraldev/dev/articles/programming_strategy.html

CAMIS - A PHUSE DVOST Working Group. Accessed on July 3,2024. https://psiaims.github.io/CAMIS

ACKNOWLEDGMENTS

Many thanks to Sanofi China and our manager, Echo Yan, for her assistance and encouragement.

RECOMMENDED READING

- Tidy design principles. Acce*ssed on July 3 2024.* https://design.tidyverse.org
- The tidyverse style guide. Acce*ssed on July 3 2024. https://style.tidyverse.org*
- Mastering Software Development in R - Roger D. Peng, Sean Kross, and Brooke Anderson. Acce*ssed on July 3 2024. https://bookdown.org/rdpeng/RProgDA*
- R Packages (2e) - Hadley Wickham & Jenny Bryan. Acce*ssed on July 3 2024. https://r-pkgs.org*
- Advanced R - Hadley Wickham. A*ccessed on July 3 2024.* https://adv-r.hadley.nz

CONTACT INFORMATION

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

En Wang
Sanofi China
6/F, HP Building, No.112, Jianguo Road, Chaoyang District, Beijing, 100022
enki.wang@sanofi.com

Longfei Li
Sanofi China
Chengdu Yintai Centre Tower 3, Tianfu Dadao Beiduan 1199, Wuhou District
Chengdu, Sichuan 610041
longfei.li@sanofi.com