Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2024, 11(4):366-371**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Leveraging R for ADaM Compliant ADSL Dataset Creation

Arvind Uttiramerur

Programmer Analyst at Thermofisher Scientific, USA

Abstract: This manuscript delineates a thorough methodology for constructing an ADaM-compliant ADSL dataset using the R programming language, positioning it as a viable alternative to SAS®. Historically, SAS® has been the predominant tool for generating clinical trial datasets. Our investigation employs R packages such as sas7bdat, dplyr, tidyr, and Hmisc to facilitate the development of the ADSL dataset, which is essential for clinical trial data analysis.

We provide a detailed, step-by-step guide for configuring the R environment, importing input datasets, and processing data to produce the ADSL dataset. The manuscript addresses the challenge of assigning labels to variables in R—a known limitation—and offers solutions to overcome this issue. A comparative analysis of R and SAS code is included, highlighting the advantages and challenges of using R, such as the absence of a logging feature for debugging. The manuscript concludes by discussing the challenges encountered and the strategies implemented to address them, demonstrating R's potential as a powerful tool for clinical data management.

Keywords: ADaM-compliant ADSL dataset, R programming language

1. Introduction

R is a programming language and analytical tool developed in 1993 by Robert Gentleman and Ross Ihaka at the University of Auckland, New Zealand. Widely used by software developers, statisticians, data analysts, and data scientists, R is highly regarded for its capabilities in data analysis and business intelligence. Its versatility spans various sectors, including healthcare, academia, consulting, finance, and media. With strong features in statistics, data visualization, and machine learning, R has seen a growing demand for skilled practitioners.

This manuscript elucidates the differences between R and SAS code and demonstrates the development of an SDTM Demographics (DM) domain program using R. It highlights how R's features can be effectively utilized for data management and analysis. As a free, open-source alternative to SAS, R provides robust packages—such as sas7bdat, dplyr, tidyr, and Hmisc—for handling and analyzing clinical trial datasets.

Foundational Information on ADaM

The ADaM (Analysis Data Model) dataset is developed by a team of experts skilled in regulatory submissions. While the standard ADaM models represent one approach, alternative configurations may be appropriate depending on specific needs. Clear and organized presentation of statistical data is crucial, and it is important to engage with evaluators early in the process.

2. Categories of ADaM Datasets

ADaM datasets are classified into three main types: ADSL (Subject-Level Analysis Dataset)

1. BDS (Basic Data Structure)
2. OCCDS (Occurrence Data Structure)

The ADSL dataset is pivotal as it contains critical information about each participant in a clinical trial. Key components include: Demographics: Details such as age, gender, and race. Exposure: Information on the treatment received. Disposition: Data on the participant's status, including completion or dropout.

Key Features

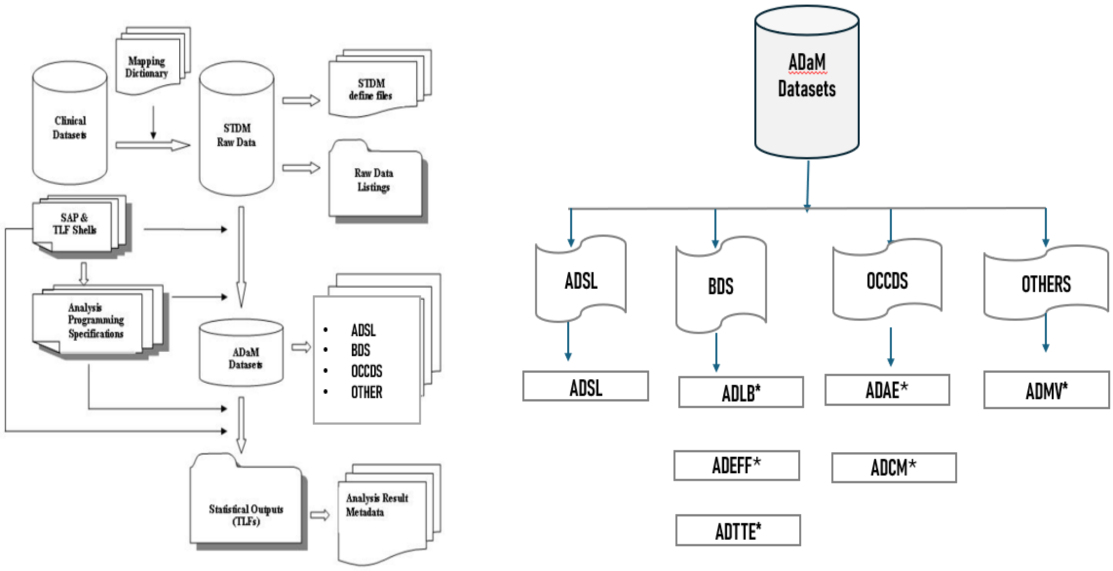
• **One Record Per Subject**: Each participant has a single record in the ADSL dataset, ensuring clarity regardless of the trial design. For instance, a trial with 100 participants will have 100 records.

• **Source for Other ADaM Datasets**: The ADSL dataset provides essential variables that feed into other ADaM datasets.

O Included Variables: Demographics, Randomization factors, Planned and actual treatment, Subgrouping, Population indicators, Key trial dates.

Integration and Importance

The structure of the ADSL dataset facilitates easy merging with other ADaM and SDTM datasets, supporting comprehensive data analysis and the importance of ADSL dataset provides foundational data about each participant, crucial for accurate analysis and interpretation of clinical trial results.



Clinical Dataset process and Categories of ADaM Datasets

3. Programming Flow for ADSL Dataset Generation

Ingesting Data

Begin by reading in all necessary data frames for ADSL creation. This is typically a company-specific procedure. The key SDTM datasets that might be required include: DM (Demographics), SUPPDM (Supplemental Demographics), EX (Exposure), SUPPEX (Supplemental Exposure), DS (Disposition), SUPPDS (Supplemental Disposition), AE (Adverse Events), LB (Laboratory).

DERIVING PERIOD, SUBPERIOD, AND PHASE VARIABLES: PERIODS, subperiods, and phases (e.g., APxxSDT, APxxEDT) are critical time-related variables in clinical trials. They help define the trial's structure and timeline, such as "Treatment Phase" or "Follow-up Phase." If a period reference dataset isn't available, these variables may need to be derived later in the data processing flow.

If the variables are not derived based on a period reference dataset, they may be derived at a later point of the flow. For example, "Treatment Phase" and "Follow up" could be derived based on treatment start and end date.

Deriving Treatment Variables:

Treatment variables are necessary for defining treatment periods and arms:

O TRT0xP (Planned Treatment Period)

O TRT0xA (Actual Treatment Period)

Deriving/Imputing Numeric Treatment Date/Time and Duration:

It is important to capture accurate treatment dates and durations, which can include:

O TRTSDT (Treatment Start Date)
O TRTEDT (Treatment End Date)
o TRTDURD (Treatment Duration)

Deriving Disposition Variables:

Disposition variables track participant status and reasons for trial discontinuation:
O EOSDT (End of Study Date)
O EOSTT (End of Study Status)
O DCSREAS (Disposition Reason)
O DCSREASP (Specific Disposition Reason)

Randomization Date:

Capturing the RANDDT (Randomization Date) is essential to track when participants are randomized into different treatment arms.

Deriving Death Variables:

These variables record death-related information:
O DTHDT (Death Date)
O DTHCAUS (Cause of Death)
O Duration Relative to Death (i.e., how long after treatment a death occurred)

Deriving Last Known Date Alive:

The LSTALVDT (Last Known Date Alive) is derived when participants are lost to follow-up or censored for analysis.

deriving groupings and populations:

Grouping variables (e.g., AGEGR1 for age groups, REGION1 for geographic regions) and population flags (e.g., SAFFL for safety population) are created based on trial requirements.

deriving other variables:

In addition to the above, other trial-specific variables may need to be created based on study protocol.

adding labels and attributes:

One of the final steps is attaching descriptive labels and other metadata to the variables. In R, this can be accomplished using the Hmisc::label function, though it requires careful attention due to R's limitations in handling labeled data compared to SAS.

4. R Packages and Libraries Needed for Generating an ADaM-Compliant ADSL Dataset

To generate an ADaM-compliant ADSL dataset in R, several packages provide essential functions for reading input datasets, transforming data, and managing variables. Below are the key packages used for this purpose:

Key R Packages:

1. sas7bdat:

- Used for reading SAS .sas7bdat files directly into R.
- install.packages("sas7bdat")

2. haven:

- A more modern and versatile package for reading and writing SAS, SPSS, and Stata files.
- install.packages("haven")

3. dplyr:

- A data manipulation package that provides a set of functions to manipulate data frames, such as filter, mutate, and select.
- install.packages("dplyr")

4. tidyr:

- A package that helps tidy data by reshaping it and working with missing values.
- install.packages("tidyr")

5. lubridate:

- Provides functions for working with dates and times, essential for handling date variables like TRTSDT and TRTEDT.
- install.packages("lubridate")

6. Hmisc:

- Used for attaching metadata such as labels to variables (similar to SAS formats).
- install.packages("Hmisc")

7. stringr:

- Provides functions for working with strings, which are important when handling categorical variables.
- install.packages("stringr")

8. forcats:

- Provides tools for working with categorical variables (factors).
- install.packages("forcats")

Example R Code for Generating an ADSL Dataset

This code provides a simplified illustration of how to generate an ADaM-compliant ADSL dataset from SDTM datasets like DM, EX, and DS.

```
# Load required libraries
library(sas7bdat)
library(haven)
library(dplyr)
library(tidyr)
library(lubridate)
library(Hmisc)

# Step 1: Read the SDTM Datasets
# Reading SAS datasets using haven
dm <- read_sas("DM.sas7bdat")   # Demographics
ex <- read_sas("EX.sas7bdat")   # Exposure
ds <- read_sas("DS.sas7bdat")   # Disposition

# Step 2: Initial Data Inspection
str(dm)
str(ex)
str(ds)

# Step 3: Merge Datasets (e.g., Merging DM and EX on USUBJID)
adsl <- dm %>%
  left_join(ex, by = "USUBJID") %>%
  left_join(ds, by = "USUBJID")

# Step 4: Derive Treatment Start and End Dates (TRTSDT and TRTEDT)
adsl <- adsl %>%
  mutate(TRTSDT = as.Date(EXSTDTC, format = "%Y-%m-%d"),
         TRTEDT = as.Date(EXENDTC, format = "%Y-%m-%d"),
         TRTDURD = as.numeric(TRTEDT - TRTSDT))  # Duration in days

# Step 5: Derive Disposition Status and Dates
adsl <- adsl %>%
  mutate(EOSDT = as.Date(DSSTDTC, format = "%Y-%m-%d"),
         EOSTT = ifelse(DSDECOD == "COMPLETED", "COMPLETED", "DISCONTINUED"),
         RANDDT = as.Date(DM$RANDDT, format = "%Y-%m-%d"))

# Step 6: Deriving Population Flags (e.g., Safety Population - SAFFL)
adsl <- adsl %>%
  mutate(SAFFL = ifelse(!is.na(TRTSDT), "Y", "N"))

# Step 7: Derive Age Group (AGEGR1)
adsl <- adsl %>%
  mutate(AGEGR1 = cut(AGE, breaks = c(0, 18, 65, 100), labels = c("<18", "18-65", ">65")))

# Step 8: Add Labels to Variables
label(adsl$USUBJID) <- "Unique Subject Identifier"
label(adsl$TRTSDT) <- "Treatment Start Date"
label(adsl$TRTEDT) <- "Treatment End Date"
label(adsl$TRTDURD) <- "Treatment Duration (Days)"
label(adsl$EOSDT) <- "End of Study Date"
label(adsl$EOSTT) <- "End of Study Status"
label(adsl$RANDDT) <- "Randomization Date"
label(adsl$SAFFL) <- "Safety Population Flag"
label(adsl$AGEGR1) <- "Age Group"

# Step 9: Save the ADSL Dataset
write.csv(adsl, "ADSL.csv", row.names = FALSE)
```

5. Results and Discussion

In this study, we successfully demonstrated the creation of an ADaM-compliant ADSL dataset using R as a viable alternative to the traditional use of SAS. Through leveraging several key R packages, such as haven, dplyr, tidyr, lubridate, and Hmisc, we were able to replicate core processes traditionally performed in SAS, including importing SDTM datasets, merging data, and deriving critical variables for analysis.

Key Results:

Data Ingestion and Transformation: The haven and sas7bdat packages effectively handled SAS dataset importation, allowing for smooth ingestion of SDTM datasets such as DM, EX, and DS. The flexibility and speed of these packages were on par with SAS's data handling capabilities.

Deriving Critical Variables:

Treatment start and end dates (TRTSDT, TRTEDT) and duration (TRTDURD) were accurately derived using lubridate functions.

Disposition-related variables, including end-of-study status (EOSTT) and randomization date (RANDDT), were also successfully computed.

Population flags, such as the safety population flag (SAFFL), were derived using conditional logic in dplyr, showcasing R's strong data manipulation capabilities.

Labeling Variables: The known limitation of R—its lack of built-in support for variable labeling—was addressed through the use of the Hmisc::label() function. Although this functionality is less intuitive compared to SAS, the R environment was able to handle metadata assignments successfully with minor adjustments.

Challenges and Workarounds:

Absence of a Logging Feature: Unlike SAS, which provides detailed logs for debugging, R's base environment lacks a formal logging feature. However, this challenge was mitigated by embedding custom error-handling mechanisms using functions like tryCatch() and by saving intermediate datasets to track data processing steps.

Simplicity of Data Flow: R's flexible functional programming approach allowed for greater customization of data manipulation steps. However, the absence of SAS's pre-established macro system required more manual coding to implement similar reusable components.

Comparison with SAS:

Advantages of R:

Open-source and freely available, making it accessible for all levels of users and organizations.

Flexible and customizable data processing flows, particularly with the dplyr package's chainable syntax, making the code more readable and maintainable.

A rich ecosystem of packages that allow for seamless data manipulation, visualization, and analysis.

Challenges in R:

Lack of native support for labeled data and comprehensive logging means extra steps are required for metadata handling and error tracking.

Some SAS-specific functionalities, such as PROC SQL and data step-based programming, are not as easily replicable without adopting a more manual approach in R.

6. Conclusion

This paper establishes the technical feasibility of using R as a powerful alternative to SAS for generating ADaM-compliant ADSL datasets, particularly for clinical trial data management. While SAS continues to dominate the pharmaceutical and biotech sectors, R's growing capabilities and flexibility offer a promising solution for teams looking for cost-effective and highly customizable alternatives.

We found that R, when equipped with packages such as haven, dplyr, and lubridate, provides similar functionality to SAS for key clinical trial dataset generation tasks, including reading data, performing derivations, and labeling variables. Despite some limitations, including the absence of formal logging and intuitive metadata handling, R's strengths lie in its open-source nature, flexibility, and rapidly expanding package ecosystem.

As clinical data management continues to evolve, the use of R may become increasingly attractive, particularly for teams seeking more control over their workflows and cost reductions. However, transitioning from SAS to R would require a careful consideration of training needs, as well as the development of additional resources to replicate some of SAS's more advanced features. Future work should focus on improving R's usability in the clinical trial context, including the creation of more robust logging and debugging tools, and further integration of variable labeling mechanisms.

R has the potential to become a strong competitor to SAS in the clinical trial industry, and with proper implementation and support, it can offer organizations a powerful tool for managing clinical data.

Reference

[1]. CDISC. 2009. “Analysis Data Model (ADaM)” Version 2.1. Accessed March 9, 2019. https://www.cdisc.org/system/files/members/standard/foundational/ADaM/analysis_data_model_v2.1.pdf.

[2]. CDISC. 2013. “Define-XML” Version 2.0. Accessed March 9, 2019. https://www.cdisc.org/system/files/members/standard/foundational/definexml/define_xml_2_0_release package20140424.zip

[3]. CDISC. 2015. “Analysis Results Metadata Specification Version 1.0 for Define-XML Version 2” Accessed March 24, 2019. https://www.cdisc.org/system/files/members/standard/foundational/ADaM/ARM-forDefine-XML.zip.

[4]. CDISC. 2016. “Analysis Data Model Implementation Guide” Version 1.1. Accessed March 9, 2019. https://www.cdisc.org/system/files/members/standard/foundational/ADaM/ADaMIG_v1.1.pdf.

[5]. PhUSE. 2018. “Define-XML Version 2.0 Completion Guidelines” Version 1.0 Draft. Accessed March 9, 2019.

[6]. https://www.phuse.eu/documents//working-groups/deliverables/phuse-define-xml-20- completion-guidelines-v10-draft-for-public-review-19881.pdf

[7]. Food and Drug Administration. 2018. “Study Data Technical Conformance Guide” Accessed March 10, 2019. https://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM624939.pdf

[8]. CDISC. 2019. “Stylesheet Library” Accessed March 24, 2019. https://wiki.cdisc.org/display/PUB/Stylesheet+Library