

PP21

Lessons learned from the ADaM data development using ADMIRAL

Katarzyna Gołasa, Tomasz Lebitko

katarzyna.golasa@astrazeneca.com, tomasz.lebitko@astrazeneca.com
Late-stage Development, Respiratory and Immunology (R&I), BioPharmaceuticals R&D, AstraZeneca, Warsaw, Poland

Abstract

The rising popularity of R in data analysis suggests a potential symbiotic relationship with SAS due to their unique strengths in analysis and reporting deliverables of Statistical Programming Departments. The ADMIRAL package has emerged as the most attractive software due to its simplicity and robust amount of CDISC oriented functions. In response, a project that involved an open-source R programming language has been designed. The group of AstraZeneca programmers decided to pioneer the implementation of R and ADMIRAL in the organization. The initiative aimed to develop various ADaM datasets in R using real study data, focusing on utilization of the ADMIRAL package and its function. We will be sharing the summary of our work, pit falls, lessons learned and pros or cons of using R for ADaM datasets development. Additionally, we will offer a detailed look at the technical side of the creation of ADaM domains in R versus SAS, including a brief overview of the most useful ADMIRAL functions

Introduction

- Workstream** – AstraZeneca workstream is a part of the cross - Therapeutic Area R initiative. The project is focused on developing the ADaM datasets using the ADMIRAL package. It is the first time R programming was used to develop ADaM datasets.
- R language** - a free software environment that provides a wide variety of statistical and graphical techniques. R can be extended via packages. Packages are the units of reproducible R code. They include reusable R functions, the documentation that describes how to use them, and sample data. The popular packages are: tidyverse, admiral, lubridate or dffdf.
- ADMIRAL*** - the purpose of ADaM in R Asset Library {admiral} is to provide an open source, modularized toolbox that enables the pharmaceutical programming community to develop ADaM datasets in R. The package consists of many useful functions prepared to answer the needs of Biometrics analysis.

Methods**

R programming	SAS programming
R is open-source software, available for free but validation of R versions and packages requires additional resources	SAS is a proprietary software, requiring a license to use
Used in academics and research	Large library of high-quality production codes
R is finding its way into commercial applications	SAS has a strong leading position in the commercial analytics space
Steeper learning curve	Shallower learning curve
Highly flexible and customizable	Less flexible with a limited library of procedures and tools
Advanced graphical capabilities (ggplot2)	Limited graphics capabilities
Different versions of R	Stable version of SAS
Packages (many versions)	No packages
FDA submissions in testing phase	Well established submissions

Results

Example 1 - Data transposition

Create an output dataset by restructuring the values in an input dataset and rearranging selected variables into observations.

R code	SAS code
dataset02<- derive_vars_transposed(dataset01, select(dataset01, USUBJID, PARAMCD, AVAL, AVISITN, ANL01FL), by_vars = exprs(USUBJID, AVISITN), key_var = PARAMCD, value_var = AVAL, filter = ANL01FL =='Y')	proc sort data=dataset01 out=dataset02; by USUBJID AVISITN; where ANL01FL="Y"; run; proc transpose data=dataset02 out=dataset02t(drop=_name_); id PARAMCD; idlabel PARAMCD; by USUBJID AVISITN; var AVAL; run; proc sort data=dataset02 nodupkey out=dataset02a; by USUBJID AVISITN; run; data dataset03; merge dataset02t(in=a) dataset02a(in=b keep=STUDYID USUBJID AVISITN AVISIT ADT ADTM ATM ADY ABLFL QSCAT AWTARGET AWTDIFF AWLO AWHI AWU ANL01FL RANDDT TRTEDT); by USUBJID AVISITN; if a; run;

Example 2 - Baseline derivation

Calculate the last known assessment prior to treatment start.

R code	SAS code
dataset02 <- dataset01 %>% restrict_derivation(derivation = derive_var_extreme_flag, args = params(by_vars = exprs(USUBJID, PARAMCD), order = exprs(ADT), new_var = ABLFL, mode = "last"), filter = (lis.na(AVAL) & ADT <= RANDDT))	proc sort data=dataset01 out=base01; by USUBJID PARAMCD ADY; where ADT <= RANDDT; run; data base02; set base01; by USUBJID PARAMCD ADY; if last.PARAMCD then ABLFL='Y'; run; data dataset02; set dataset01 (where=(ADT > RANDDT)) base02; run;

Example 3 - Date & Time imputation

Replace missing or partially missing date/time entries in a dataset with substituted values.

R code	SAS code
*** Date imputation dataset02 <- dataset01 %>% derive_vars_dt(dtc = MHENDT, new_vars_prefix = "AEND", highest_imputation = "M", date_imputation = "last")	data dataset02; set dataset01; if length(MHENDTC) eq 7 then do; mhendtc1=input(MHENDTC, anydtdte7.); lastday=intnx ('month',mhendtc1,0,'E'); mhendtc2=put(lastday, is8601da.); end; else if length(MHENDTC) eq 4 then do; mhendtc2 = MHENDTC substr(mhendtc2,5,6)="-12-31"; end;
*** Datetime imputation dataset02 <- dataset01 %>% derive_vars_dt(dtc = MHENDT, new_vars_prefix = "AEND", highest_imputation = "h", date_imputation = "first", time_imputation = "first")	if MHENDTC ne "" and length(MHENDTC) eq 10 then mhendt = input(MHENDTC,yymmdd10.); else if mhendtc2 ne "" then mhendt = input(mhendtc2,yymmdd10.); format mhendt yymmdd10.; run;

Example 4 - Merge datasets

Combine two or more datasets based on shared variables.

R code	SAS code
dataset02 <- derive_vars_merged(dataset01, dataset_add = sv, by_vars = exprs(USUBJID), new_vars = exprs(SVSTDTC,VISITNUM))	proc sort data = dataset01; by USUBJID; run; proc sort data = dataset02; by USUBJID; run; data dataset03; merge dataset01(in=a) dataset02(keep=USUBJID SVSTDTC VISITNUM); by USUBJID; if a; run;

Example 5 - Other useful functions

Other useful examples of R codes.

Function	R code	SAS code
Compare datasets	diffdf(dataset01, dataset02)	proc compare base = dataset01 compare = dataset02 listall warning listobs listvar method = absolute criterion = .00001; id STUDYID USUBJID; run;

Conclusions

- The SAS statistical programmers community has a vast amount of clinical sources to learn from. The R statistical programmers community is growing in number but current R users will need some time to develop its functionalities in clinical research.
- ADMIRAL package, which is under constant development, gives access to the CDISC-oriented functions that may support users in code production. SAS does not contain such a variety of functions and requires the development of macros.
- R packages may be amended and the functions that were used in the past may be removed or moved to other packages. This makes reusing R code a challenge due to an update in R version or package. This problem is not present in SAS.
- SAS submissions are well established, whereas the R-based test submission package has been conducted by the R consortium and FDA but it is not a standard business approach yet.

Acknowledgments

We would like to thank the AstraZeneca R Biometrics Leadership Team for sponsoring and supporting the R ADMIRAL initiative and R Biometrics Workstream.

(*) <https://github.com/pharmaverse/admiral>
(**) <https://support.sas.com/en/documentation.html>
<https://pharmaverse.org/>
<https://www.lexjansen.com/>
<https://www.cdisc.org/>