

CDISC - SDTM/ADaM Dataset Generation Using R

Prasanna Murugesan, AstraZeneca Pharmaceuticals, Gaithersburg, MD, USA

Introduction

Open source statistical software R is being used in several industries for data analysis and data visualizations to provide insights. Although, R has been used in exploratory analysis in Pharma/Biotech industry for a long time, it has not been used for creating and analyzing clinical trial datasets (SDTM/ADaM). Traditionally, SAS® has been used to generate clinical trial datasets. In this poster we will explore the feasibility of generating SDTM/ADaM datasets using base, dplyr, haven and lubridate packages of R.

The Rise of R

- Open source statistical computing software
- Widely used in exploratory analysis and visualizations
- Most preferred among academic institutions
- Popularity increased with big data and data analytics
- Used for analyzing pre-clinical, POPPK/PD, non clinical data (Post marketing data / HEOR)

R for CDISC - SDTM/ADaM creation

R packages for Data Transformation

- Base : Base R functions
- Dplyr : designed for data transformation
- Lubridate : Date functions
- Haven : Loads and exports SAS® files

SAS® vs R-dplyr syntax

SAS®	R-dplyr
keep/drop	select
data step derivations	mutate
by statement	group_by
if/where statement	filter
proc sort	arrange
statistical procedures (means/tabulate)	summarise
transpose	spread/gather
merge	left/right/inner/full joins
set by rows/columns	bind_cols/bind_rows

Libname setup equivalent in R- Sample code

```
sdtm <- "///folderlocation/data/sdtm"
out <- "C:/"
```

Read files

```
dm <- read_sas(file.path(sdtm,"dm.sas7bdat"))
ds <- read_sas(file.path(sdtm,"ds.sas7bdat"))
sv <- read_sas(file.path(sdtm,"sv.sas7bdat"))
suppsv <- read_sas(file.path(sdtm,"suppsv.sas7bdat"))
```

Subject level derivation -Sample code

```
adsl <- dm %>% # read %>% as "and then"
  select(subjid, age, sex, ht, wt, race, scrfl) %>%
  mutate(bmi = (weight*703)/height^2 ) %>%
  filter(scrfl == "Y") %>%
  select(-scrfl) %>%
  arrange(subjid)
```

Merging parent and supplemental dataset

```
suppds_ <-suppds %>%
  mutate(idvarval_ = as.numeric(idvarval)) %>%
  select(usubjid,idvarval_,qnam,qval) %>%
  spread(.,qnam,qval)
ds_all <-left_join(ds, suppds_, by =
  c("usubjid"="usubjid","dsseq"="idvarval_"))
```

Baseline/change from baseline derivation

```
advs <-advs_ %>%
  group_by(subjid,paramcd) %>%
  arrange(subjid,paramcd) %>%
  mutate ( base = aval[visitnum==1],
    ablfl = ifelse(visitnum == 1,"y",na),
    chg = ifelse ( !is.na(aval) & !is.na(base),aval-base,na),
    pchg= ifelse ( !is.na(aval) & !is.na(base),((aval-
      base)/base)*100,na))
```

Benefits

- An independent tool to validate SAS® datasets
- Cost effective open source tool
- More statistical functionalities available in R
- Datasets viewed in R-Shiny make it more user friendly

Challenges

- Frequent version changes in R packages
- Packages may not have gone through SDLC
- Dataset names/vars & syntax in R are case sensitive
- Logs are not detail oriented as SAS®

Reference:

- <http://dplyr.tidyverse.org/>
- https://rpubs.com/bradleyboehmke/data_wrangling
- <https://www.rstudio.com/> - data wrangling cheat sheet