

## Exploring AE with EDC Data Using R Package

Feng Chen, Fosunpharma

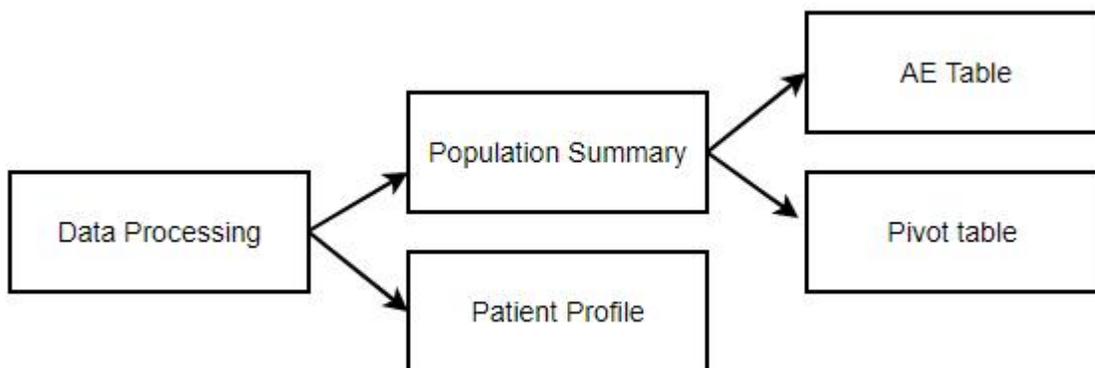
### ABSTRACT

Adverse events (AE) are a major component in safety evaluation, where comes the demand of efficiency in generating outputs. However, the conventional tabulation process based on Analysis Data Model (ADaM) datasets can be time-consuming for routine monitoring. In this paper, we explored a way to summarize AE data via an R Shiny App starting from raw AE dataset and a Study Data Tabulation Model (SDTM) – like dataset of Demographics (DM).

### INTRODUCTION

R Shiny, with its various packages, serves as an alternative to generated Table, Figure, Listings (TFLs) generated in terms of clinical data demonstration. It can be especially helpful for medical monitoring before any official dry-run is accomplished. Therefore, we came to the question whether an App could be built based on EDC data to save time from creating standard SDTM or ADaM datasets. Considering that among regular outputs contents of AE are relatively fixed despite difference in study designs, we chose AE as our target to perform data visualization.

### APP STRUCTURE



**Figure 1. Flowchart of building a raw AE-based Shiny App**

The app contains a global file, an UI, a server and several modules. In the global file, we conducted all the data processing work. Data derived or processed would then be transferred to the major UI and server where modules were called. It consists of two functional panels - population summary and patient profile. The former focuses on tabulation of quantitative data and the latter helps to explore data on an individual level.

### DATA PREPARATION

Before we could obtain any meaningful results from our Shiny app, derivation for standardized variables was necessary. First, we need a ADSL-like dataset to provide information on core variables, which consist of TRT01A, TRT01AN, TRTSDT, TRTEDT, DTHDT, SAFFL. To obtain date format variables, we started from character variables in EDC whose year, month and day parts were identified by a regex pattern and were combined to form a date in YYMMDD10 format. The derivation of these variables depended largely on study design and Statistical Analysis Plan (SAP) and the related codes might hence lack reusability. Next, we would focus on raw AE and create a pseudo-ADAE. The majority of variables required little processing, such as releveling the original AEREL. However, to derive TRTEMFL, ASTDT

and AENDT should trigger attention. We could choose to derive ASTDT and AENDT only from complete raw dates and decide treatment emergent using date of first dose and 30 days after last dose as the conventional threshold. Otherwise, we could stick to SAP to impute missing values and find other applicable threshold. This choice affected the accuracy of outputs and thus should be made based on the App's usage scenario.

Below code shows our process to derive AENDT according to study SAP:

```
ae_rawdate <- list("STD" = "AESTDAT_RAW",
                  "END" = "AEENDAT_RAW")

date_pattern <- "(\\w{2,4})?.+?(\\w{1,4})?.+?(\\w{1,4})?"

ae %<>% within(., {
  year <- stri_match(get(ae_rawdate$END), regex=date_pattern,
                    cg_missing = "")[,2]
  month <- stri_match(get(ae_rawdate$END), regex=date_pattern,
                     cg_missing = "")[,3]
  day <- stri_match(get(ae_rawdate$END), regex=date_pattern,
                    cg_missing = "")[,4]
  year_y <- grepl(pattern="\\d+", year)
  month_y <- grepl(pattern="\\d+", month)
  day_y <- grepl(pattern="\\d+", day)
  AEENDTC <- case_when(
    year_y & month_y & day_y ~ ymd(paste(sep="-", year, month, day)),
    year_y & month_y & !day_y ~ ceiling_date(ymd(paste(
      sep="-", year, month, "01")), "month")-1,
    year_y & !month_y & !day_y ~ ymd(paste(
      sep="-", year, "12-31"))
  )
  AEENDT <- convert_dtc_to_dt(as.character(AEENDTC))
  AEENDTC <- ifelse(!is.na(DTHDT) & DTHDT<AEENDT, ymd(DTHDT), AEENDTC)
  AEENDTC <- as.Date(AEENDTC, origin=lubridate::origin)
  AENDT <- convert_dtc_to_dt(as.character(AEENDTC))
}
```

After producing ready-to-use datasets, we moved on to prepare data and values for our display content. To improve the reusability of codes and to facilitate its potential functioning across multiple studies, we avoided directly using variable names in function arguments or assigning certain values when we created our UI and server. For example, in Display 1. UI for Population Summary, the rows displayed and the corresponding filtering condition were predefined in the global file instead of module files. With the aid of R package **rlang**, strings can be evaluated as arguments or variable names in our codes. In this way, it will consume less effort to apply the apply to another study.

## POPULATION SUMMARY

**Select Treatment Groups:**

All

**Count by:**

Patients

Events

Adverse Events Coded Using  
MedDRA 24.0

Summary of TEAE
AE by groups
Pivot Table

**Event Filter:**

	DRUG 100mg	DRUG 400mg	Total
TEAE	18 (100.0)	8 (100.0)	26 (100.0)
CTCAE≥3级	9 (50.0)	8 (100.0)	17 (65.4)
与研究药物相关	18 (100.0)	8 (100.0)	26 (100.0)
导致剂量调整	6 (33.3)	7 (87.5)	13 (50.0)
导致暂停用药	0 (0.0)	0 (0.0)	0 (0.0)
导致永久停药	0 (0.0)	0 (0.0)	0 (0.0)
严重不良事件	0 (0.0)	0 (0.0)	0 (0.0)
导致死亡	0 (0.0)	0 (0.0)	0 (0.0)

Previous 1 Next

Showing 1 to 8 of 8 entries

### Display 1. UI for Population Summary

This tab panel includes routine table contents. Its first section is a summary of Treatment Emergent Adverse Event (TEAE) where a user can choose the treatment groups displayed and to count by patients or events. The grouping variable derives from TRT01A and its level was prespecified during data preparation to display in an appropriate order. The Next section contains tables of TEAEs by System Organ Class, Preferred Term and TEAEs by System Organ Class and Preferred Term by maximum toxicity grade. A sidebar on the left side and a drop-down checkbox floating above the table are available for the user to filter records, such as choosing a subset with maximum toxicity grade grouped by System Organ Class. The third section enables the user to design a pivot table by dragging and dropping with the help of R package **sortable** to build up our UI. In the previous sections, tabulation is performed on predefined groups, but in this section any character variable can serve as a grouping variable as long as it exists in our ADSL-like dataset or ADAE-like dataset created beforehand. Plus, after deciding the columns and rows the user can click on a blue button to add filter boxes, where available options derive from the two datasets. The difference is that date and numeric variables are also available to work as filters.

Summary of TEAE

AE by groups

Pivot Table

Variables

Row

实际分组描述

国家

AETERMSOC

SAFFL

TRTEMFL

AETERMPT

SAEYN

AECTC

AECTC

AETERM

Column

AREL1

Show 10 entries Search: 

AETERM	相关	不相关
ALT增高	1	0
AST增高	0	1
GGT增高	2	0
LDH增高	0	1
T波改变	1	1
三尖瓣疾病	0	1
丘疹脓疱性皮炎	1	0
丙氨酸氨基转移酶(谷丙转氨酶)增高	1	0
丙氨酸氨基转移酶(谷丙转氨酶)增高	1	0
丙氨酸氨基转移酶升高	1	1

Previous 1 2 3 4 5 ... 13

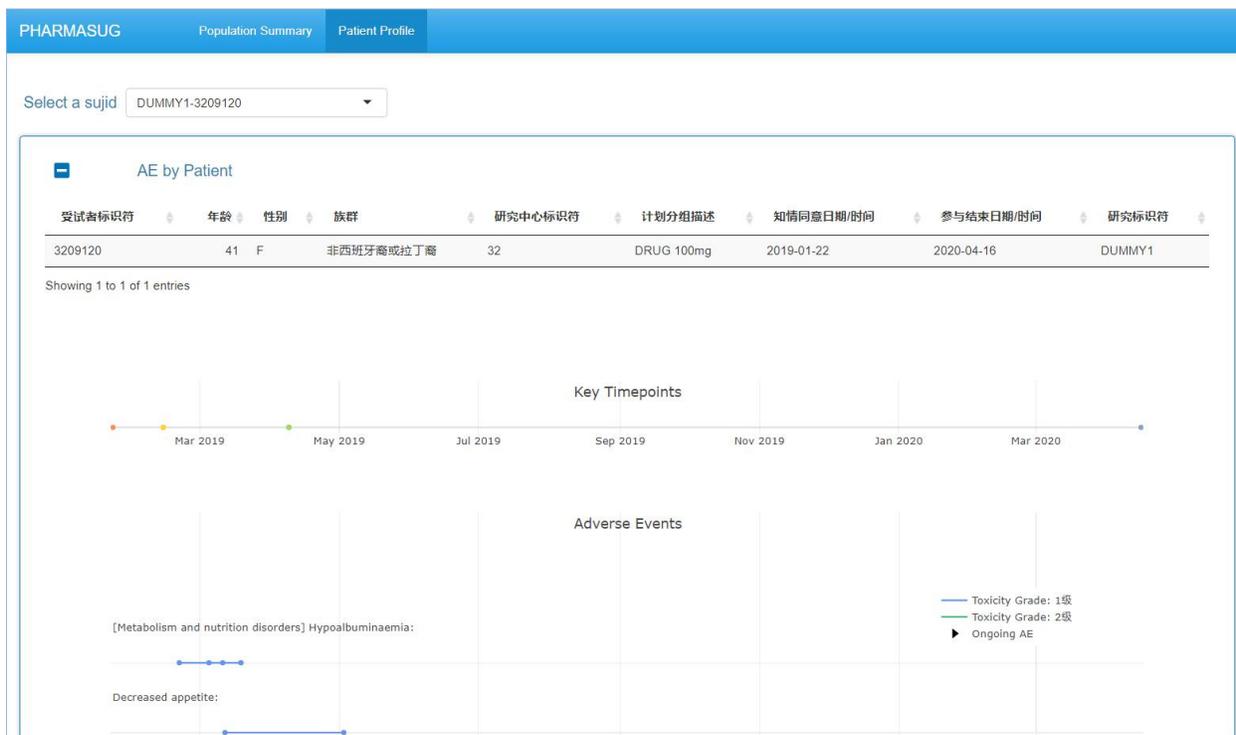
Next

≡ ADD A FILTER

## Display 2. A Pivot Table Grouped by AREL1

### PATIENT PROFILE

This tab panel visualizes the AE profile of any single patient, where each line segment stands an AE reported term or dictionary-derived term, the horizontal axis standing for start or end date and the colors representing the Common Terminology Criteria for Adverse Events (CTCAE) grading. The user can tell the duration and severity of AE from the plot and if he or she places the mouse pointer on any line segment, hovering text will show details on the corresponding AE. Besides, a table follows the line plot will further inform the App user of any findings about the patient. When a user clicks on a line segment, the table will display raw AE data of the selected reported term or dictionary-derived term that the line segment represents. He or she can choose to view another related raw dataset, including Laboratory and Electrocardiograph.



**Display 3. Patient Profile of a Selected ID**

## CONCLUSION

Building up a Shiny App to tabulate and visualize AE based on EDC data is feasible and has the following strengths.

- No need to derive all standard SDTM and ADaM variables
- Traceability of raw values without imputation

However, there are still obstacles to be solved.

- High demand on quality of EDC data, such as proper labeling
- Need to weigh the balance between accuracy and efficiency

## ACKNOWLEDGMENTS

I would like to acknowledge my friend, Happy, for keeping me company and making a lot of noise when I was coding.

## CONTACT INFORMATION

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

Feng Chen

FosunPharma

cf1996224@163.com

Any brand and product names are trademarks of their respective companies.