**PharmaSUG China 2018 – Paper AD-44**

# A Macro to Automate RFPENDTC Derivation

Dan LI, Sanofi-Aventis, Chengdu, Sichuan, China

## ABSTRACT

RFPENDTC is a date/time variable of Demography domain when a subject ended participation or follow-up in a trial and it equals to the last known date for this subject. A common idea for RFPENDTC derivation is to find out the latest date for all domain datasets in the database of a subject. Searching for each date/time variable of all domain datasets manually is not only tedious but also prone to errors owing to omissions. Rapidly growing demands towards working hour efficiency also puts against handwritten code for RFPENDTC derivation. A macro for RFPENDTC derivation automation is proposed consequently in this paper, which avoids the problem that manuscript code is easy to cause omission and improves coding experience by reusing code.

## INTRODUCTION

RFPENDTC is the date/time when a subject ended participation or follow-up in a trial and it equals to the last known date for this subject. From this definition, a common idea of derivation is to find out the value of all date/time variables of a subject in each domain dataset in the database, and then the maximum of these values is the subject's RFPENDTC. Repeat this process for other subjects then RFPENDTC for all subjects will be obtained.

However, applying this method could cause several inconveniences. First, manually implementing the above process requires copying the names of all domains and the names of all time variables in each domain. It is easy to miss some domains or some variables. Second, such code is difficult to reuse and modify.

The SAS macro facility provides a degree of encapsulation of the code. Well-designed macro parameters make the code's applicable conditions clear at a glance. By re-encapsulating the code blocks inside the macro into macros, the logic level of each functional module is made clear by the nesting of macros. Coupled with more comprehensive documentation and comments, macro users can apply the logic of the macro without recall the whole process.

Taking advantage of SASHELP.VCOLUMN, a macro to automatically derive RFPENDTC is proposed. Using the data set name of SASHELP.VCOLUMN and the variable name of the data set, the time variable of all domain data sets is automatically selected and the maximum value is obtained. The final data set also includes source variables that RFPENDTC comes from.

## WHAT THE MACRO DOES

The main function of this macro is to find out the RFPENDTC of each subject and show which variable(s) this date comes from. With optional parameters, the user can also specify which domain data sets do not participate in the search process, and how the macro recognizes whether a variable is a date/time.

The above functions are reflected in the macro parameters, and the macro parameter list is explained in Table 1:

| Parameter | Meaning | Mandatory | Default |
|---|---|---|---|
| in_lib | The logical library in which the dataset to be searched is located | No | The logical library name of the SDTM datasets (e.g. RDBS) |
| exclude_domain | Specify which domain datasets are not involved in the search. If there are multiple domain datasets, separate by commas (e.g. %str(vs, dm, pr)) | No | - |
| date_var_str | Character combination that a date variable must contain. If the combination is more than one, separate them by commas (e.g. %str(dtm, dt)) | No | DTC |
| data_out | Name of the output dataset | Yes | - |

**Table 1. Macro parameters**

Based on the above parameters, the macro user can search through datasets using both basic mode and advanced mode. In basic mode, the macro deems a variable a date/time variable if it contains DTC in the variable name, and no dataset is excluded. The user only needs to specify the name of the output dataset. Macro call in basic mode is shown as below:

```
%rfpendtc(data_out=result);
```

In advanced mode, the user needs to specify the rule that the macro recognize a date/time variable. For example, date_var_str=%str(dt, dtm) means if a variable name contains dt or dtm then it is a date/time variable. Macro call in advance mode is shown as below:

```
%rfpendtc(
    in_lib         =my_lib
    ,data_out      =result
    ,date_var_str  =%str(dt, dtm)
    ,exclude_domain=%str(dm, pr, vs, se)
);
```

## HOW THE MACRO DOES THE JOB

### OPEN SASHELP.VCOLUMN WITH CONDITIONS

SASHELP.VCOLUMN is a dictionary view provided to the user. It contains all logic library names, all dataset names and all variable names in the current SAS session. A subset of SASHELP.VCOLUMN table is shown in Table 2, where LIBNAME is the logical library name, MEMNAME the data set name, and NAME the variable name:

| Libname | Memname | Name |
|---|---|---|
| RDBS | AE | AESTDTC |
| RDBS | AE | AEENDTC |
| RDBS | VS | VSDTC |
| … | … | … |

**Table 2. Part of SASHELP.VCOLUMN**

Each observation in SASHELP.VCOLUMN contains the name of the dataset, the name of the logical library it is in, and the name of the variable in the dataset. Forming a conditioning statement that holds the macro parameter date_var_str, you can open SASHELP.VCOLUMN and remain only the observations of a date/time variable in the opened dataset. By iterating through this dataset, the date/time variable name and its value are fetched. Each time a date/time variable and its value are fetched, they will be appended to a pooled dataset. After the iteration terminates, the pooled dataset will contain all date/time variables and values without omission.

Codes below illustrate how to iterate through SASHELP.VCOLUMN and pool date/time variables together.

First create a template dataset, so that after each sub-dataset is obtained, it can be appended to the template dataset to pool together.

```
data temp;
    format USUBJID RPF SRCVAR $100. DOMAIN $4. RPF_T datetime20.;
run;
```

Then construct a processing statement that deals with the date variable string entered by the user:

```
%let dtcnt = %sysfunc(countw(&date_var_str, &delims));
%do i=1 %to &dtcnt;
    %let dtc_statement_temp = %upcase(%sysfunc(scan(&date_var_str, &i, &delims)));
    %let dtc_statement = &dtc_statement or index(name, "&dtc_statement_temp.") > 0;
%end;
%let statement = %sysfunc(catt(where=(libname="&in_lib" and
(&dtc_statement))))%nrstr(;);
```

Constructs a statement that specifies a domain dataset that does not participate in the search:

```
%let excnt = %sysfunc(countw(&exclude_domain, &delims));
%do i=1 %to &excnt.;
    %let exclude_domain_str = %upcase(&exclude_domain_str)
%upcase(%sysfunc(scan(&exclude_domain, &i, &delims)));
%end;
```

Open SASHELP.VCOLUMN with constructed statement that &statement represents:

```
%let dsid = %sysfunc(open(sashelp.vcolumn(&statement)));
```

For example, if the macro parameter date_var_str has a value of %str(dt, dtm), then the statement above equals to:

```
%let dsid = %sysfunc(open(sashelp.vcolumn(
    where=(libname="&in_lib" and (index(name, "dt") > 0 or index(name, "dtm") > 0))
)));
```

Thus SASHELP.VCOLUMN is opened with date and domain conditions specified by macro parameter date_var_str and exclude_domain.

## ITERATE THE DATASET OF DSID AND POOL TOGETHER ALL DATE VALUES

```
%syscall set(dsid);
%let index = 0;
%do %while(%sysfunc(fetch(&dsid)) EQ 0);
    %let dt = %sysfunc(reverse(&memname.));
    %let dt2 = %sysfunc(substr(&dt, 3));
    %let domain = %sysfunc(reverse(&dt2));
    %if NOT (%sysfunc(index(&exclude_domain_str, &domain)) GT 0) %then %do;
        %let index = %eval(&index + 1);
        data temp_add&index.(keep=usubjid rpf srcvar domain rpf_t);
            format usubjid rpf srcvar $100. domain $4. rpf_t datetime20.;
            set &in_lib..&memname.;
            where trim(&name.) NE "";
            srcvar = "&name";
            rpf = &name.;
            if trim(usubjid) EQ "" then delete;
        run;
        %impute_date(data_in=temp_add&index, data_out=temp_add&index._t,
date_var=rpf, temp_var=rpf_t);
        proc append base=temp data=temp_add&index._t;run;
    %end;
%end;
```

In the above code block, lines 4-6 first get the name of the data set domain. Because in the study I am currently coping with, the data set naming rules are domain name + underscore + H, so you can use this method to get the name of the domain. Line 7 first determines whether the domain is in the exclude domain list specified by parameter exclude_domain. If not, subsequent process will be performed. Lines 9 through 16 construct the corresponding sub-dataset using the dataset name and variable name of the current fetch. Line 18 appends the sub-dataset of each loop construct to the template dataset. When the loop ends, a dataset containing each and all date/time variables will be pooled together. The function of the macro at line 17 is to impute the value of the time variable of various lengths into the is8601 format and then convert it to a numeric value for comparison. The function of this code block is shown below in Figure 1:

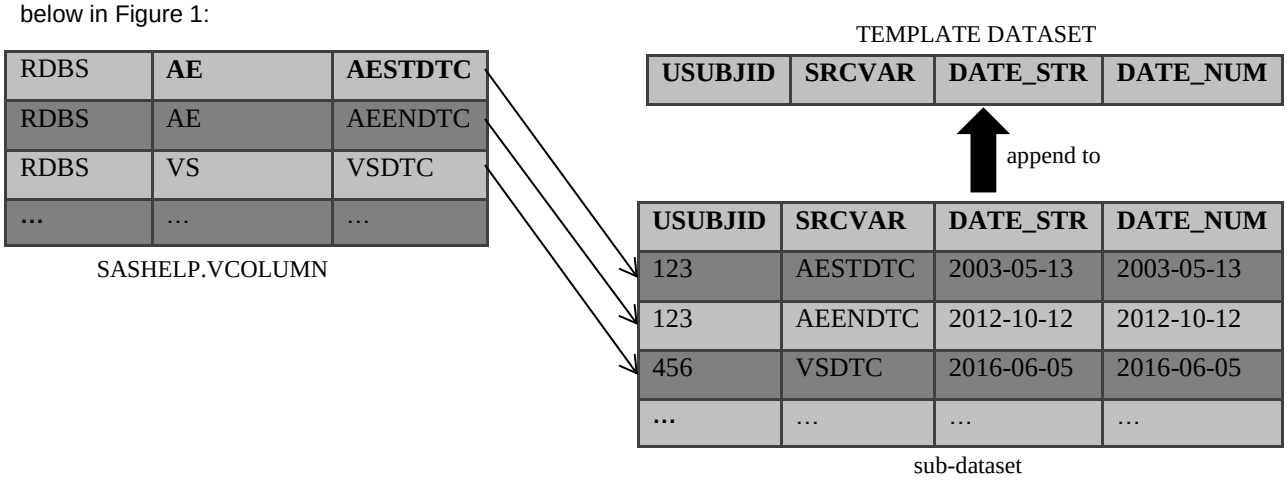


**Figure 1. Core idea table view**

### FIND OUT THE OBSERVATION WHICH HAS THE MAXIMUM NUMERIC DATE VALUE OF A SUBJECT

```
proc sql;
    create table temp_t as
    select usubjid, rpf as rfpendtc, max(rpf_t) as rfpendtc_t, srcvar
    from temp
    group by usubjid
    having max(rpf_t) = rpf_t;
quit;
data &data_out.(drop=rfpendtc_t srcvar);
    set temp_tt;
    retain USUBJID RFPENDTC SOURCE;
    length source $200;
    by usubjid rfpendtc_t srcvar;
    retain source;
    if first.rfpendtc_t then source = srcvar;
    else source = compress(source || "," || srcvar);
    if last.usubjid;
run;
```

In the pooled dataset, find the maximum date_num (date/time value in numeric format) of each subject, and output the result dataset, as shown in Figure 2 below:

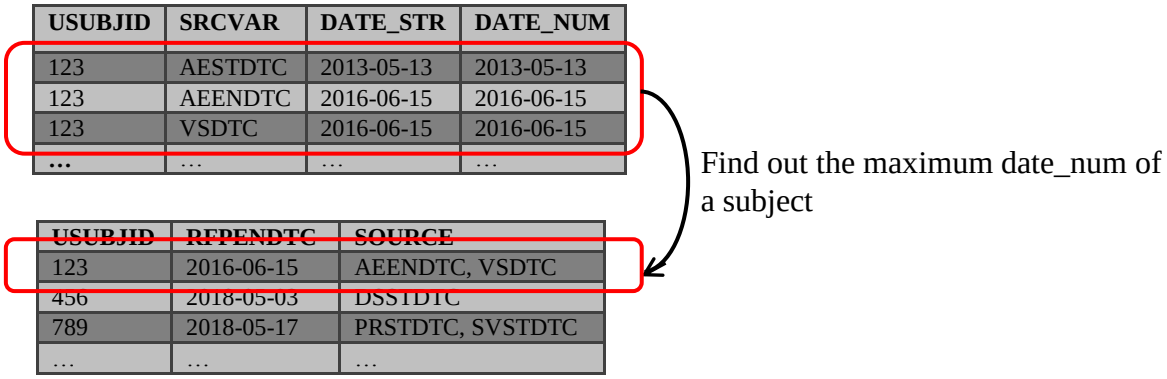


**Figure 2. Find the maximum date value of a subject**

## CONCLUSION

The macro proposed automates the tiring process of derivation of RFPENDTC, and avoided the error caused by omitting any date variables. The user of this macro could customize derivation process by specify macro parameters date_var_str and exclude_domain.

## REFERENCES

Study Data Tabulation Model Implementation Guide: Human Clinical Trials Version 3.2

## CONTACT INFORMATION

Name: Dan LI
Enterprise: Sanofi-Aventis
Address: 39F, in99 Chengdu Yintai Center, 1199 North Section of Tianfu Road, High-Tech Zone, Chengdu, P.R.C.
Email: dan8.li@sanofi.com