



Paper CT01

A Macro to Automate Detection Of Date Variables and Derivation Of RFPENDTC.

P.Kiran Kumar babu, Quartesian, Chennai. India.

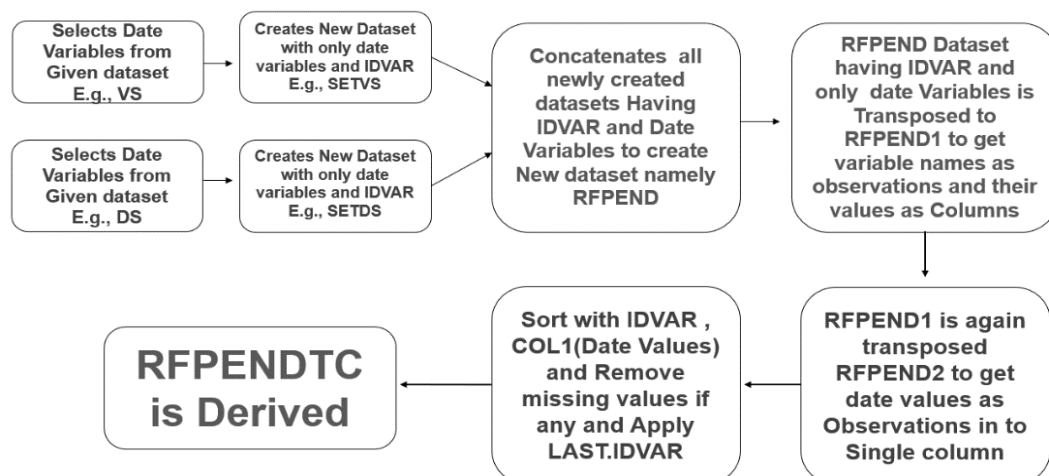
ABSTRACT

RFPENDTC is a Demographic variable, As per the Implementation Guide, RFPENDTC is the Date/time when a subject ended participation or follow-up in a trial, as defined in the protocol. It should correspond to the last known date of contact. Examples include completion date, withdrawal date, last follow-up, date recorded for lost to follow-up, or death date. So, it requires a check of various sdtm domains, depending up on the study for last contact date of a subject. Manually searching each domain and date variables for latest dates requires lot of time limiting working efficiency and also results in very long code. A simple and easy to understand macro for RFPENDTC derivation using transpose and last.variable is proposed consequently in this paper.

INTRODUCTION

RFPENDTC represents the end of the subject's involvement in the study. In general, it should correspond to the last known date of contact. Although there are some examples where RFPENDTC may be the last date among all records for the subject in the database. In any case, End of Participation date should be filled if a patient has ended the study. Derivation of RFPENDTC involves a process that requires date variables from different datasets as per the study. Manually identifying date variables and finding latest date requires lot of time and also prone to errors as it is easy to miss date variables by mistake and also it is difficult to modify or reuse manual codes.

SCHEMATIC REPRESENTATION : HOW IT WORKS



UNDERSTANDING THE MACRO WITH HELP OF DUMMY STUDY:

Example: VS and DS datasets from dummy study

1. Selects date variables as per required date format using proc contents as shown in Table 1.1.

```
proc contents data=vs noprint out=vars
  (where=(format="MMDDYY"));
run;
```

identifies variable names with required formats.

i.e. VSDT VSDT1 VSDT3 VSDT4 as shown in Table 1.1.

	LIBNAME	MEMNAME	NAME	FORMAT
1	RAW	VS	VSDT	MMDDYY
2	RAW	VS	VSDT1	MMDDYY
3	RAW	VS	VSDT3	MMDDYY
4	RAW	VS	VSDT4	MMDDYY

Table 1.1

2. Create new dataset with prefix "SET" (for easy concatenation) with

IDVAR and only date variables as in Table 1.2.

IDVAR is nothing but any variable representing Subject.

E.g. : Subjectid, Usubjid, Screen no etc.

```
data setvs;
  set vs(encoding=any);
  idvar=subjectid;
  KEEP IDVAR VSDT VSDT1 VSDT3 VSDT4;
run;
```

Prefix "SET" can be changed as per user wish, but it is essential
To keep unique prefix for easy concatenation.

IDVAR	VSDT	VSDT1	VSDT3	VSDT4
101	03/16/2017	.	.	.
101	.	.	.	.
101	.	03/21/2017	.	.
101	.	03/21/2017	.	.
101	.	03/21/2017	.	.
101	.	03/21/2017	.	.
101	.	03/21/2017	.	.
101	.	03/22/2017	.	.
101	.	03/22/2017	.	.
101	.	03/22/2017	.	.
101	.	03/22/2017	.	.
101	.	03/22/2017	.	.
101	.	03/22/2017	.	.
101	.	03/23/2017	.	.
101	.	.	03/23/2017	03/23/2017

Table 1.2

3. Similarly steps were performed with DS dataset and output is given as in Table 1.3 and Table 1.4.

```
proc contents data=DS noprint out=vars
  (where=(format="MMDDYY"));
run;
```

```
data setds;
  set ds(encoding=any);
  idvar=subjectid;
  KEEP IDVAR DSDT;
run;
```

LIBNAME	MEMNAME	NAME	FORMAT
RAW	DS	DSDT	MMDDYY

Table 1.3

IDVAR	DSDT
101	03/31/2017
102	03/31/2017
103	03/31/2017
104	04/07/2017
105	04/07/2017
106	04/07/2017

Table 1.4

3. Concatenates datasets SETDS and SETVS datasets by IDVAR with date variables to develop a new dataset **RFPEND** as in table 1.5.

```
data rfpend;
set setvs setds;
by idvar;
run;

proc sort data=rfpend;
by idvar;
run;
```

Though we have all dates required for analysis
It is not possible to sort the data horizontally and
get latest date as dates are present in different
observations , so its essential to Reshape the data
for easy sorting.

IDVAR	DSDT	VSDT	VSDT1	VSDT3	VSDT4
101	03/31/2017	.	.	.	.
101	.	03/16/2017	.	.	.
101	.	.	.	.	.
101	.	.	03/21/2017	.	.
101	.	.	03/21/2017	.	.
101	.	.	03/21/2017	.	.
101	.	.	03/21/2017	.	.
101	.	.	03/21/2017	.	.
101	.	.	03/22/2017	.	.
101	.	.	03/22/2017	.	.
101	.	.	03/22/2017	.	.
101	.	.	03/22/2017	.	.
101	.	.	03/22/2017	.	.
101	.	.	03/22/2017	.	.
101	.	.	03/23/2017	.	.
101	.	.	.	03/23/2017	03/23/2017
101	.	.	.	03/23/2017	.
101	.	.	.	03/23/2017	.
101	.	.	.	.	.
101	.	.	.	03/23/2017	.
101	.	.	.	.	.
101	.	.	.	.	.
102	03/31/2017	.	.	.	.
102	.	03/16/2017	.	.	.

Table 1.5

4. Dataset RFPEND is transposed to RFPEND1 with date variables in **VAR** statement to get date values and output is as shown in Table 1.6.

```
proc transpose data=rfpend out=rfpend1(rename=( _name_ =name));
by idvar;
var VSDT1 VSDT2 VSDT3 VSDT4 DSDT;
run;
```

IDVAR	NAME	_LABEL_	COL1	COL2	COL16	COL17	COL23
101	DSDT	Date of Subject Complete or Discontinue the Study	03/31/2017	.	.	.	.
101	VSDT	Vital Signs Date of Assessment	.	03/16/2017	.	.	.
101	VSDT1	Vital Signs Date of Assessment 1	.	.	.	.	.
101	VSDT3	Vital Signs Date of Assessment 3	.	.	03/23/2017	03/23/2017	.
101	VSDT4	Vital Signs Date of Assessment 4	.	.	03/23/2017	.	.
102	DSDT	Date of Subject Complete or Discontinue the Study	03/31/2017	.	.	.	.
102	VSDT	Vital Signs Date of Assessment	.	03/16/2017	.	.	.
102	VSDT1	Vital Signs Date of Assessment 1	.	.	.	.	.
102	VSDT3	Vital Signs Date of Assessment 3	.	.	03/23/2017	03/23/2017	.
102	VSDT4	Vital Signs Date of Assessment 4	.	.	03/23/2017	.	.
103	DSDT	Date of Subject Complete or Discontinue	03/31/2017	.	.	.	.

Table 1.6

5. Dataset rfpnd1 is transposed to rfpnd2 with all **COL** variables in **VAR** statement to get date values in single column for each subject as shown in Table 1.7.

```
proc sort data=rfpnd1;
by idvar name;
run;

proc transpose data=rfpnd1 out=rfpnd2;
by idvar name;
var col;;
run;

/*sorting datasets to remove missing values*/
proc sort data=rfpnd2;
by idvar coll;
where coll ne .;
run;
```

IDVAR	NAME	_NAME_	COL1
101	VSDT	COL2	03/16/2017
101	VSDT1	COL4	03/21/2017
101	VSDT1	COL5	03/21/2017
101	VSDT1	COL6	03/21/2017
101	VSDT1	COL7	03/21/2017
101	VSDT1	COL8	03/21/2017
101	VSDT1	COL9	03/22/2017
101	VSDT1	COL10	03/22/2017
101	VSDT1	COL11	03/22/2017
101	VSDT1	COL12	03/22/2017
101	VSDT1	COL13	03/22/2017
101	VSDT1	COL14	03/22/2017
101	VSDT1	COL15	03/23/2017
101	VSDT3	COL16	03/23/2017
101	VSDT3	COL17	03/23/2017
101	VSDT3	COL18	03/23/2017
101	VSDT3	COL20	03/23/2017
101	VSDT4	COL16	03/23/2017
101	DSDT	COL1	03/31/2017
102	VSDT	COL2	03/16/2017

Table 1.7

All COL's are based on visits of subject, but we need only latest date.

6. In the last step ,after sorting rfpnd2 ,derive latest date from dataset using last.idvar as shown in Table 1.8.

```
data rfpndtc;
set rfpnd2;
by idvar coll;
if last.idvar;
rfpndtc=put(coll,is8601da.);
drop _name_;
run;
```

Final dataset : RFPENDTC

IDVAR	NAME	RFPENDTC
101	DSDT	2017-03-31
102	DSDT	2017-03-31
103	DSDT	2017-03-31
104	DSDT	2017-04-07
105	DSDT	2017-04-07
106	DSDT	2017-04-07
107	DSDT	2017-04-07
108	DSDT	2017-04-19
109	DSDT	2017-04-19
110	DSDT	2017-04-18

Table 1.8



MACRO PARAMETERS AND LIMITATIONS:

The main function of this proposed macro is to find out the RFPENDTC of each subject, Macro parameters such as libname, **idvar**(identification variable) and dsnames must be addressed i.e., to use this macro user should specify library and which domain datasets that should participate in the search process, also the required date format should be addressed in UPPERCASE inside quotations without length.

Macro Limitation mainly includes IDVAR, it must be the same variable from all domains(E.g. If Subject id is considered as IDVAR then Subject id should be present in all domains used in search process and must be same in all datasets. With optional parameters, the user can specify the format of dates to consider for the last participation. user can always modify proposed macro based on requirement.

Method : 2 as proposed in appendix requires user to drop few variables to get exact results ,since method 2 pulls up all dates including system generated date variables .

Method : 3 In case of dates separated by year ,month and day user can concatenate dates prior to creation of Rfpend and where clause has to be updated as per user requirement as proposed in Method 3 in appendix .

CONCLUSION :

The proposed macro with exact known date format automates the derivation of RFPENDTC, and avoided the error caused by omitting any date variables. The user of this macro can restrict this macro up to creation of rfpend dataset with idvar and only date variables ,and transposing can be done by manually choosing variables required for analysis. Also, user can get date variables with help of sashelp.vcolumn or dictionary.columns directly but it is advisable to create individual dataset for each domain with their date variables as they can be helpful to perform QC effectively. The user of this macro can always update the macro to address their requirement as data is not always constant.

REFERENCES

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

ACKNOWLEDGMENTS

I would like to thank Mr. Koteswar Govind, Mr. Antony Tharcis Mariaselvaraj, Mr. Kannan Murugaiyan, Ms. Ramya Sengodan, Mr. Suresh Periyasamy and Mr. Rajesh Edara for their Support and encouragement.

CONTACT INFORMATION

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

Author Name: P Kiran Kumar Babu

Company: Quartesian Research Private Limited.

Address : Chennai ,India

Work email:kiran.babu@quartesian.com

Brand and product names are trademarks of their respective companies.



June 14th–18th

US Connect 2021

The Clinical Data
Science Conference

Appendix:

```
%macro rfpndtc (libname=work,dsnames=,idvar=subjectid,format='MMDDYY',dropvar=,out=rfpndtc);
```

```
/*Create Domain Datasets with Only Date Variables******/
```

```
  %do i=1 %to %sysfunc(countw(&dsnames))
    %let newdsn=%scan(%upcase(&dsnames),&i,%str( ));
    %let lib=%upcase(&libname);
```

```
/*Using proc contents to filter date variables with required format*/
```

```
proc contents data=&lib..&newdsn(encoding=any)noprint out=vars (where=(format=&format));
run;
```

```
/*Method :2- User can modify where statement in proc contents as per user requirement*/
```

```
/* (where=(index(name,"DT")>0 or index(name,"DAT")>0)); OR (where=(format ne " "));*/
```

```
*Check no of observations in VARS to find datasets with missing date variables;
```

```
proc sql noprint;
  select nobs into :nobs separated by ' ' from dictionary.tables
  where libname='WORK' and memname='VARS';
quit;
```

```
%if &nobs>0 %then %do;
```

```
/*creating macro variable VAR with required variables for further use*/
```

```
proc sql noprint;
  select distinct name into: vars separated by ' '
  from vars;
quit;
```

```
/*creating new dataset with Prefix " SET" for easy appending in later steps*/
```

```
data set&newdsn;
  set &lib..&newdsn(encoding=any);
  idvar=&idvar;
  keep idvar &vars;
```

```
run;
```

```
proc sort data=set&newdsn;
  by idvar;
```

```
run; %end;
```

```
%else %do;
```

```
  %put WARNING "NO DATE VARIABLES in &NEWDSN dataset";
  %end;
```

```
%end;
```

```
*Selects all the datasets starting with name "set";
```

```
proc sql noprint;
  select memname into :names separated by ' '
  from dictionary.tables
  where libname='WORK' and memname like "SET%";
```



June 14th–18th

US Connect 2021

The Clinical Data
Science Conference

```
quit;
*Set all datasets to rfpnd dataset;
data rfpnd(drop=&dropvar);
  set &names;
  by idvar;
run;
*Creating macro variable Transvars for transpose variables in rfpnd datasets;
proc sql noprint;
  select distinct name into: Transvars separated by ' '
  from dictionary. Columns
  where libname="WORK" and
  memname="RFPEND" and name NOT IN("IDVAR");
quit;

proc sort data=rfpnd;
  by idvar;
run;
/*Transpose data to move all date variables in to one variable */
proc transpose data=rfpnd out=rfpnd1(rename=( _name_ =name));
  by idvar;
  var &Transvars;
run;
proc sort data=rfpnd1;
  by idvar name;
run;
/*Transpose data to move all visit dates in to one variable */
proc transpose data=rfpnd1 out=rfpnd2;
  by idvar name;
  var col:;
run;

/*sorting datasets to remove missing values*/
proc sort data=rfpnd2;
  by idvar col1;
  where col1 ne .;
run;
*For last visit in each domain for each subject;
data &out;
set rfpnd2;
by idvar col1;
if last.idvar;
  rfpndtc=put(col1,is8601da.);
drop _name_;
run;
%mend;
%rfpndtc(libname=raw,dsnames=ae vs,dropvar=tpdt);
```



June 14th–18th

US Connect 2021

The Clinical Data
Science Conference

Method 3:

```
%macro rfpndtc (libname=work,dsnames=,idvar=id);

%do i=1 %to %sysfunc(countw(&dsnames));
  %let newdsn=%scan(%upcase(&dsnames),&i,%str( ));
  %let lib=%upcase(&libname);

proc contents data=&lib..&newdsn(encoding=any) noprint out=vars
(where=(index(name,"DAT_YYYY")>0 OR index(name,"DAT_MM")>0 OR index(name,"DAT_DD")>0));
run;

/*creating macro variable VAR with required variables for further use with keywords*/
proc sql noprint;
  select distinct name, name into: numnames separated by ",",
                  : numdates separated by ',best.))||"-||strip(put('

  from vars
  where index(name,"ST")=0;
quit;
%if %length(&numnames)ne 0 %then %do;
data set&newdsn;
set &lib..&newdsn(encoding=any);

idvar=&idvar;
if nmiss(&numnames)=0 then do;
  &newdsn.dates=strip(put(&numdates,best.));end;
keep idvar &newdsn.dates;
run;

proc sort data=set&newdsn;
by idvar;
run;
%end;
%else %do;
  %put WARNING "NO VARIABLES WITH KEYWORD "YYYY or MM or DD" in &NEWDSN dataset";
%end;
%end;

*Selects all the datasets starting with name "set";
proc sql noprint;
  select memname into :names separated by ' '
  from dictionary.tables
  where libname="WORK" and memname like "SET%";
quit;
*Set all datasets to rfpnd dataset;
data rfpnd;
set &names;
by idvar;
run;

*****THE END*****
```