



QUARTESIAN

PHUSE US CONNECT 2021

CT01

A Macro to automate detection of date variables and derivation of **RFPENDTC**

P.Kiran Kumar babu

QUARTESIAN

AGENDA :

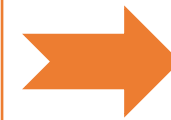
- Working process of macro
- Identification of date variables
- Combining datasets
- Reshaping of Data
- Getting latest date
- Macro parameters
- Macro limitations
- Advantages of macro
- Conclusion

WORKING PROCESS

**Select date
variables
from desired
datasets**



**Reshaping of
data**



**Get latest
date.**

Identification of Date Variables:

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

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

Table 1.1

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

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
101	.	.	03/23/2017	.

- Prefix can be of user discrete : SET

Dataset : SETVS

Identification of Date Variables:

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

LIBNAME	MEMNAME	NAME	FORMAT
RAW	DS	DSDT	MMDDYY

Table 1.2

- Most of the data management software's captures dates of study in different format w.r.t system generated date variables

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

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

Dataset : SETDS

Identification of Date Variables:

```
proc contents data=VS(encoding=any) noprint out=vars  
(where=(format ne " "));  
run;
```

LIBNAME	MEMNAME	NAME	FORMAT
DUMMY	VS	FormEntryDate	DATETIME
DUMMY	VS	VS DT	MMDDYY
DUMMY	VS	VS DT1	MMDDYY
DUMMY	VS	VS DT3	MMDDYY
DUMMY	VS	VS DT4	MMDDYY
DUMMY	VS	Vital_Sign_CREATE_DTM	DATETIME
DUMMY	VS	Vital_Sign_LAST_UPD_DTM	DATETIME

Table 1.3

```
proc contents data=VS(encoding=any) noprint out=vars  
(where=(index(name,"DT")>0));  
run;
```

LIBNAME	MEMNAME	NAME	FORMAT
DUMMY	VS	VS DT	MMDDYY
DUMMY	VS	VS DT1	MMDDYY
DUMMY	VS	VS DT3	MMDDYY
DUMMY	VS	VS DT4	MMDDYY
DUMMY	VS	Vital_Sign_CREATE_DTM	DATETIME
DUMMY	VS	Vital_Sign_LAST_UPD_DTM	DATETIME

Table 1.4

- These methods gives additional date variables and are not very accurate , but additional variables can be dropped

Identification of Date Variables:

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

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	.	.

Dataset : SETVS

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

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

Dataset : SETDS

Combining datasets:

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

Dataset : RFPEND

- User can modify macro to select date variables by using **SASHELP.VCOLUMN** or **DICTIONARY.COLUMNS** , but it is advisable to create individual dataset for each domain with only date variables and idvar
- Reshaping of data is required

	IDVAR	DSDT	VSDT	VSDT1	VSDT3	VSDT4
1	101	03/31/2017	.	.	.	.
2	101	.	03/16/2017	.	.	.
3	101	.	.	.	.	.
4	101	.	.	03/21/2017	.	.
5	101	.	.	03/21/2017	.	.
6	101	.	.	03/21/2017	.	.
7	101	.	.	03/21/2017	.	.
8	101	.	.	03/21/2017	.	.
9	101	.	.	03/22/2017	.	.
10	101	.	.	03/22/2017	.	.
11	101	.	.	03/22/2017	.	.
12	101	.	.	03/22/2017	.	.
13	101	.	.	03/22/2017	.	.
14	101	.	.	03/22/2017	.	.
15	101	.	.	03/23/2017	.	.
16	101	.	.	.	03/23/2017	03/23/2017
17	101	.	.	.	03/23/2017	.
18	101	.	.	.	03/23/2017	.
19	101	.	.	.	.	.
20	101	.	.	.	03/23/2017	.
21	101	.	.	.	.	.
22	101	.	.	.	.	.
23	102	03/31/2017	.	.	.	.
24	102	.	03/16/2017	.	.	.
25	102	.	.	.	.	.
26	102	.	.	03/21/2017	.	.

Transposing of data:

```
proc transpose data=rfpnd out=rfpnd1(rename=(_name_=name));  
by idvar;  
var VSDT1 VSDT2 VSDT3 VSDT4 DSDT;  
run;
```

	IDVAR	NAME	COL1	COL2	COL4	COL5	COL6
1	101	DSDT	03/31/2017	.	.	.	.
2	101	VSDT	.	03/16/2017	.	.	.
3	101	VSDT1	.	.	03/21/2017	03/21/2017	03/21/2017
4	101	VSDT3	.	.	.	.	.
5	101	VSDT4	.	.	.	.	.
6	102	DSDT	03/31/2017	.	.	.	.
7	102	VSDT	.	03/16/2017	.	.	.
8	102	VSDT1	.	.	03/21/2017	03/21/2017	03/21/2017
9	102	VSDT3	.	.	.	.	.
10	102	VSDT4	.	.	.	.	.

Dataset : RFPEND1

Transposing of data:

```
proc sort data=rfpending1;  
by idvar name;  
run;
```

```
proc transpose data=rfpending1 out=rfpending2;  
by idvar name;  
var col;;  
run;
```

```
/*sorting datasets to remove missing values*/
```

```
proc sort data=rfpending2;  
by idvar col1;  
where col1 ne .;  
run;
```

	IDVAR	NAME	_NAME_	COL1
1	101	VSDT	COL2	03/16/2017
2	101	VSDT1	COL4	03/21/2017
3	101	VSDT1	COL5	03/21/2017
4	101	VSDT1	COL6	03/21/2017
5	101	VSDT1	COL7	03/21/2017
6	101	VSDT1	COL8	03/21/2017
7	101	VSDT1	COL9	03/22/2017
8	101	VSDT1	COL10	03/22/2017
9	101	VSDT1	COL11	03/22/2017
10	101	VSDT1	COL12	03/22/2017
11	101	VSDT1	COL13	03/22/2017
12	101	VSDT1	COL14	03/22/2017
13	101	VSDT1	COL15	03/23/2017
14	101	VSDT3	COL16	03/23/2017
15	101	VSDT3	COL17	03/23/2017
16	101	VSDT3	COL18	03/23/2017
17	101	VSDT3	COL20	03/23/2017
18	101	VSDT4	COL16	03/23/2017
19	101	DSDT	COL1	03/31/2017
20	102	VSDT	COL2	03/16/2017
21	102	VSDT1	COL4	03/21/2017
22	102	VSDT1	COL5	03/21/2017

Dataset : RFPEND2

Last.variable:

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

FINAL DATASET : RFPENDTC

	IDVAR	NAME	RFPENDTC
1	101	DSDT	2017-03-31
2	102	DSDT	2017-03-31
3	103	DSDT	2017-03-31
4	104	DSDT	2017-04-07
5	105	DSDT	2017-04-07
6	106	DSDT	2017-04-07
7	107	DSDT	2017-04-07
8	108	DSDT	2017-04-19
9	109	DSDT	2017-04-19
10	110	DSDT	2017-04-18

Macro code:

```
%rfpndtc(libname=work,dsnames=vs ds,format="MMDDYY",idvar=subjectid  
out=rfpndtc);
```



QUARTESIAN

```
%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;

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

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

%if &nobobs>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;

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%";
quit;

*Set all datasets to rfpnd dataset;
data rfpnd(drop=&dropvar);
set &names;
by idvar;
run;

*creating macro variable nvars for transpose variables in rfpnd datasets;
proc sql noprint;
select distinct name into:nvars 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 &nvars;
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;

*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;

/*sorting datasets
proc sort data=rfpnd2;
by idvar col1;
where col1 ne .;
run;
```

Macro Parameters:

Parameter	Parameter-Explanation	Optional	Default
libname	Name of a library in which desired datasets are present Example: raw , work	No	Not applicable
dsnames	Space separated list of data set names to be selected from the given library Example: dsnames=ae vs lb	No	Not applicable
idvar	variable representing subject id in the study Example: idvar=usubjid idvar="STUDYID" '-' SITE '-' SUBJECTID	No	SUBJECTID
format	Check with desired dateformat and should be given in UPPER CASE in quotes without length. Example: Format="DATETIME" OR "MMDDYY"	No	MMDDYY
dropvar	Variables that are not required for analysis , can be used only after running once and knowing variable names	Yes	
out	Out dataset name as needed	Yes	RFPENDTC

Macro Limitation:

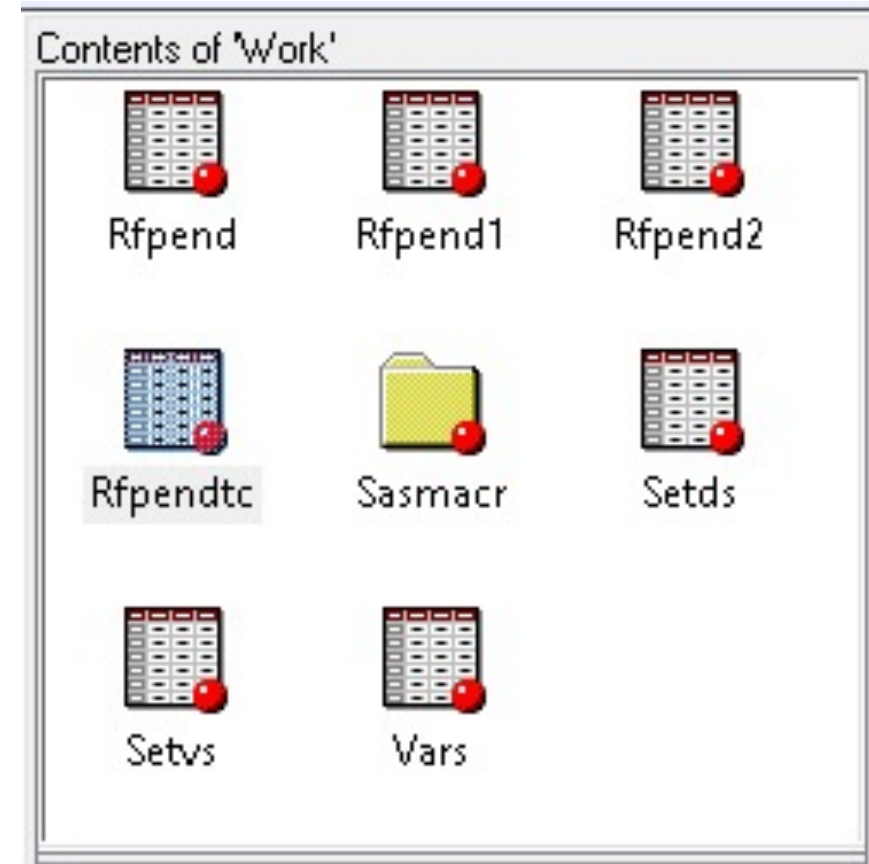
- Macro limitation includes **IDVAR** i.e., Subject identification variable , it must be present in all datasets and as same in all datasets **E.g., SUBJECTID ..**
- This macro very effective when given specific format.
- User may get undesired results if format is not specific , but later user can drop those variables which are not needed for analysis with help of macro variable **&dropvar** by manually analyzing

RFPEND dataset

E.g., %rfpendtc(libname=work,dsnames=vs ds,format="MMDDYY",idvar=subjectid,
dropvar=vsdt vsdt1,out=rfpendtc);

Advantages :

- Each domain creates a new dataset with only date variables ,very effective to perform QC
- Easy to understand and can be modified as per user needs
- Creates limited number of datasets therefore execution is fast
- Very effective against SDTM final datasets



Conclusion:

- An automated macro for Rfpendtc targeting date formats, really saves lot of programmer's time and also gives accuracy
- This Macro can be limited up to formation of RFPEND dataset, and transposing can be done manually by choosing the variables to participate in rfpndtc derivation at discrete of user
- Identification of dates based on formats and transposing them is not only useful for rfpndtc , but also useful for indexing of dates as per domains and also to perform any other actions related to dates
- Incase of dates separated by year ,month and day or separate timing variable, user can concatenate them while creating individual dataset.
- This macro can be customized as per user requirements

Questions ?

Thank you

Author:

P. Kiran Kumar babu

Statistical programmer I

kiran.babu@quartesian.com