Converting Non-Imputed Dates (for SDTM Data Sets) With PROC FCMP

Noory Kim

17 October 2016

cros nt

The Center of Excellence for Clinical Trial Data

SDTM data sets require ISO8601 date formats.

Raw data sets may have non-ISO8601 formats

- *DDMMMYYYY*
- *MM-DD-YYYY*

SDTM* data sets require ISO8601 formats

- *YYYY-MM-DD*
- *YYYY-MM*
- *YYYY*

*Study Data Tabulation Model

Source: http://xkcd.com/1179/

cros nt The Center of Excellence for Clinical Trial Data

Target output of the function to be defined

Input	Target Output	Rule
17OCT2016	2016-10-17	
17Oct2016	2016-10-17	Ignore case for month
UNOCT2016	2016-10	
UNUNK2016	2016	
17UNK2016	2016	Omit day if month unknown
17OCTUKUK	*null*	Omit day and month if year unknown

PROC FCMP functions are user-defined functions.

Once a PROC FCMP function is defined, compiled, and located, it can be invoked just like a built-in SAS function.

```
data two;
   set one;
   length date_iso8601 $10;
   date_iso8601 = convertdate(date_date9);
run;
```

PROC FCMP syntax is similar to DATA step syntax.

Example: To avoid the truncation of values of character variables, use LENGTH statements (just as you would in a DATA step).

FCMP cannot use everything a DATA step can, including

- the IN operator
- the ?? format modifier

Output 1: The Main Example

The NOTDIGIT() function detects missingness codes having at least one non-numeric character.

Obs	date_ date9	date_ iso8601
1	17OCT2016	2016-10-17
2	17Oct2016	2016-10-17
3	UNOCT2016	2016-10
4	UNUNK2016	2016
5	17UNK2016	2016
6	17OCTUNKN	
7	99JAN2016	2016-01-99
8	31FEB2016	2016-02-31

Output 1: The Main Example

However, the NOTDIGIT() function is not sufficient to detect missingness codes that are strictly numerical.

Obs	date_ date9	date_ iso8601
1	17OCT2016	2016-10-17
2	17Oct2016	2016-10-17
3	UNOCT2016	2016-10
4	UNUNK2016	2016
5	17UNK2016	2016
6	17OCTUNKN	
7	99JAN2016	2016-01-99
8	31FEB2016	2016-02-31

Output 2: The Main Example, Modified

The paper discusses two methods for preventing nonexistent dates.

Obs	date_ date9	date_ iso8601
1	01JAN2016	2016-01-01
2	01jan2016	2016-01-01
3	17OCT2016	2016-10-17
4	UNOCT2016	2016-10
5	UNUNK2016	2016
6	UNUNKUKUK	
7	17UNK2016	2016
8	17OCTUKUK	
9	99JAN2016	**2016-01**
10	31FEB2016	**2016-02**

Preventing Non-existent Dates

Method 1: Compare the date with the last existing date of the same month

- use the INTNX() function

```
year = input(yyyy, 8.);
month = input(mm, 8.);
day = input(dd, 8.);

month_start_date = mdy(month, 1, year);
month_end_date   = intnx('month', month_start_date, 0, 'end');
month_end_day    = day(month_end_date);

if day > month_end_day then outdate = yyyy || '-' || strip(mm);
else outdate = yyyy || '-' || strip(mm) || '-' || dd;
```

Preventing Non-existent Dates

Method 2: Check if date can be converted to a non-missing numeric date value

- use the ANTDTDTE10. format

```
outdate = yyyy || '-' || strip(mm) || '-' || dd;

outdate_numeric = input(outdate, anydtdte10.);
if outdate_numeric < .z then outdate = yyyy || '-' || strip(mm);
```

Selected References

Carpenter, Arthur L. “Using PROC FCMP to the Fullest: Getting Started and Doing More.” PharmaSUG 2013.

Adams, John H. “The new SAS 9.2 FCMP Procedure, what functions are in your future?” PharmaSUG 2010.

Fan, Jueru. “Trivial Date Tasks? PROC FCMP Can Help.” PharmaSUG 2016.