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

Noory Kim
Statistical Programmer, CROS NT LLC, Chapel Hill NC USA
MS Biostatistics, UNC Chapel Hill
SAS Certified Advanced Programmer for SAS 9

PharmaSUG 2017
Paper TT07

Section 1: Defining Custom Functions with the SAS Function Compiler (PROC FCMP)

Why use PROC FCMP?

- To write custom functions for use *inside* a DATA or PROC step statement

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

- To increase the modularity and reusability of your code

Example

```
proc fcmp outlib=funccol.functions.conversions;
     function ToCelsius(fahrenheit);
          celsius = 100/180 * (fahrenheit-32);
          return(celsius);
     endsub;
run;
...
data temperatures;
     set sashelp.humid;
     BulbTempC = ToCelsius(BulbTemp);
     AirTempC = ToCelsius(AirTemp);
run;
```

- This FCMP step defines and compiles a function.

- This DATA step invokes the function (twice).

Skeleton of a PROC FCMP step

```
proc fcmp outlib=libname.dataset.package;
     function functionName(inputVar1 <$>,…) <$>;
          < function code >
          return(outputVar);
     endsub;
run;

options cmplib=(libname.dataset);
```

- $: input var is char
- $: output var is char

- Loads compiled functions for use in the current SAS session.

Syntax of a PROC FCMP step

- PROC FCMP syntax is similar to DATA step syntax.
- Example: Need to use LENGTH statements to avoid outputting truncated character values.

```
proc fcmp outlib=libname.dataset.package;
    function functionName(inputVar1 <$>,…) <$>;
        length outputVar $10;
        < function code >
        return(outputVar);
    endsub;
run;
```

Syntax of a PROC FCMP step

- Caveat: Not all DATA step syntax is compatible with PROC FCMP.
  - IN operator
  - ?? format modifier

Section 2: Converting Dates for SDTM Datasets

Date Values in SDTM Data Sets

- Use ISO8601 date formats
  - YYYY-MM-DD
  - YYYY-MM
  - YYYY
- FDA on imputing dates
  - SDTM data sets: Partial dates cannot be imputed*
  - ADaM data sets: Imputation of partial dates allowed

*That is, missing components cannot be guesstimated.

Paper Example: Expected Input

- All input date values are 9 characters following the pattern *DDMMMYYYY*

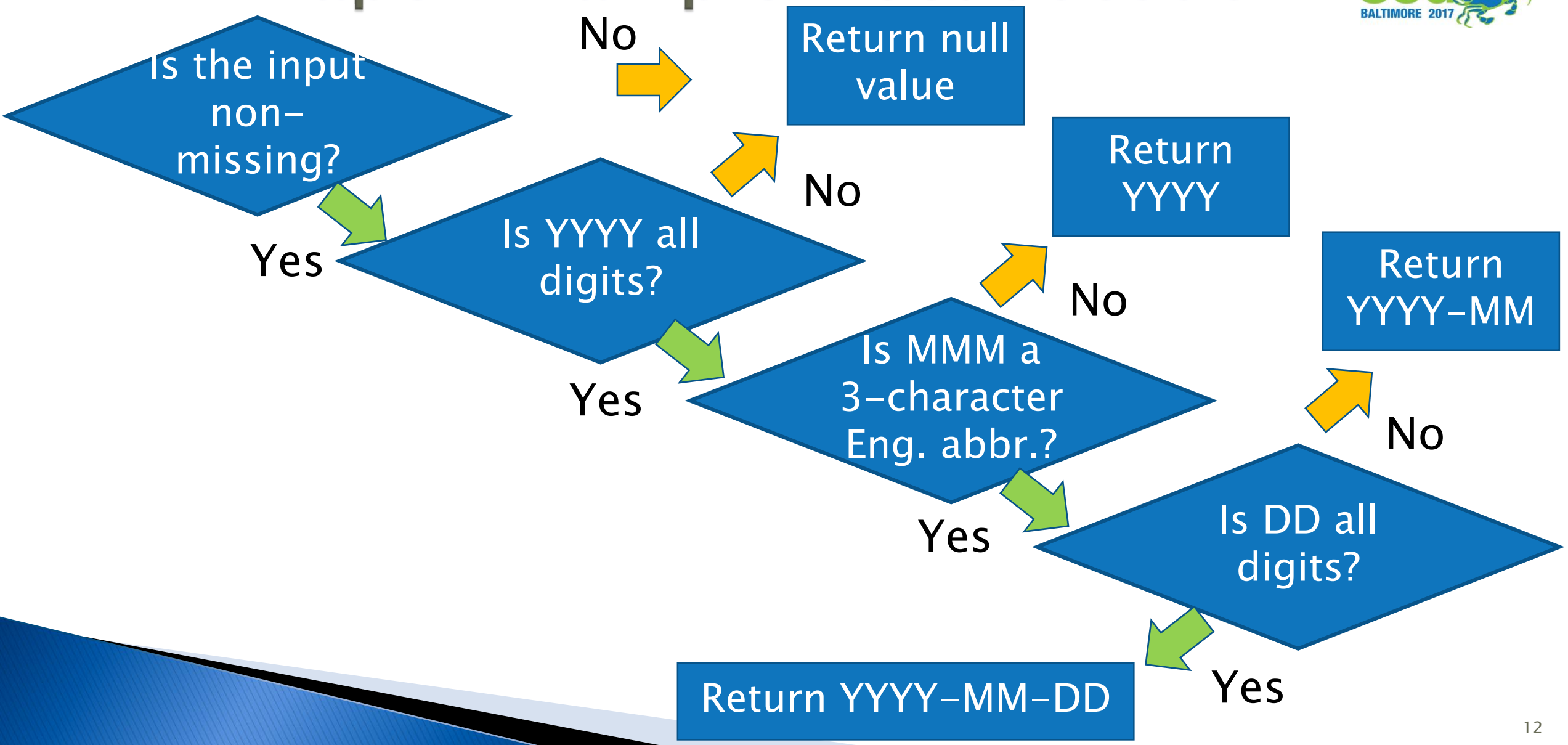
- Any year or day with a non-numeric character will be regarded as missing.
  - **UK**MAY2017 → 2017-05
  - 14MAY**UNKN** → *null*

- Any month not matching an English abbreviation (“JAN”, “FEB”, etc.) will be regarded as missing.
  - 14**UNK**2017 → 2017

Paper Example: Target Output

Expected Input	Target Output	Conversion Rule
14MAY2017	2017-05-14	
14May2017	2017-05-14	Ignore letter case of month component
UNMAY2017	2017-05	
UNUNK2017	2017	
14UNK2017	2017	Ignore day value if month is unknown
14MAYUKUK	*null*	Ignore day and month if year is unknown
UNUNKUKUK	*null*	
99JAN2017	2017-01	Ignore day if after last actual day of the month
31FEB2017	2017-02	Ignore day if after last actual day of the month

Paper Example: Pseudocode



Paper Example: Layer 1 – Year

```
function convertDate(indate $) $;
     length outdate $10;
     if length(indate) ne ' ' then do;
          yyyy = substr(indate, 6, 4);
          if notdigit(yyyy) = 0 then do;

               ... <Layers 2 and 3>

          end;
          else outdate = ' ';
     else outdate = ' ';
endsub;
```

Is the input non-missing?

Is YYYY all digits?

Return null value

Paper Example: Layer 2 – Month

```
mmm = upcase(substr(indate, 3, 3));
mm = put(mmm, $month.);

if mm ne '  ' then do;

      ... <Layer 3>

end;
else outdate = yyyy;
```

Is MMM a 3-character Eng. abbr.?

Return YYYY

- In this example the format $month uses standard English abbrevations (e.g. ‘JAN’ = ‘01’; ‘FEB’ = ‘02’, …, other = ‘ ’)

Paper Example: Layer 3 – Day

```
dd = substr(indate, 1, 2);

if notdigit(dd) = 0 then do;

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

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

Is DD all digits?

Return YYYY-MM-DD

Return YYYY-MM

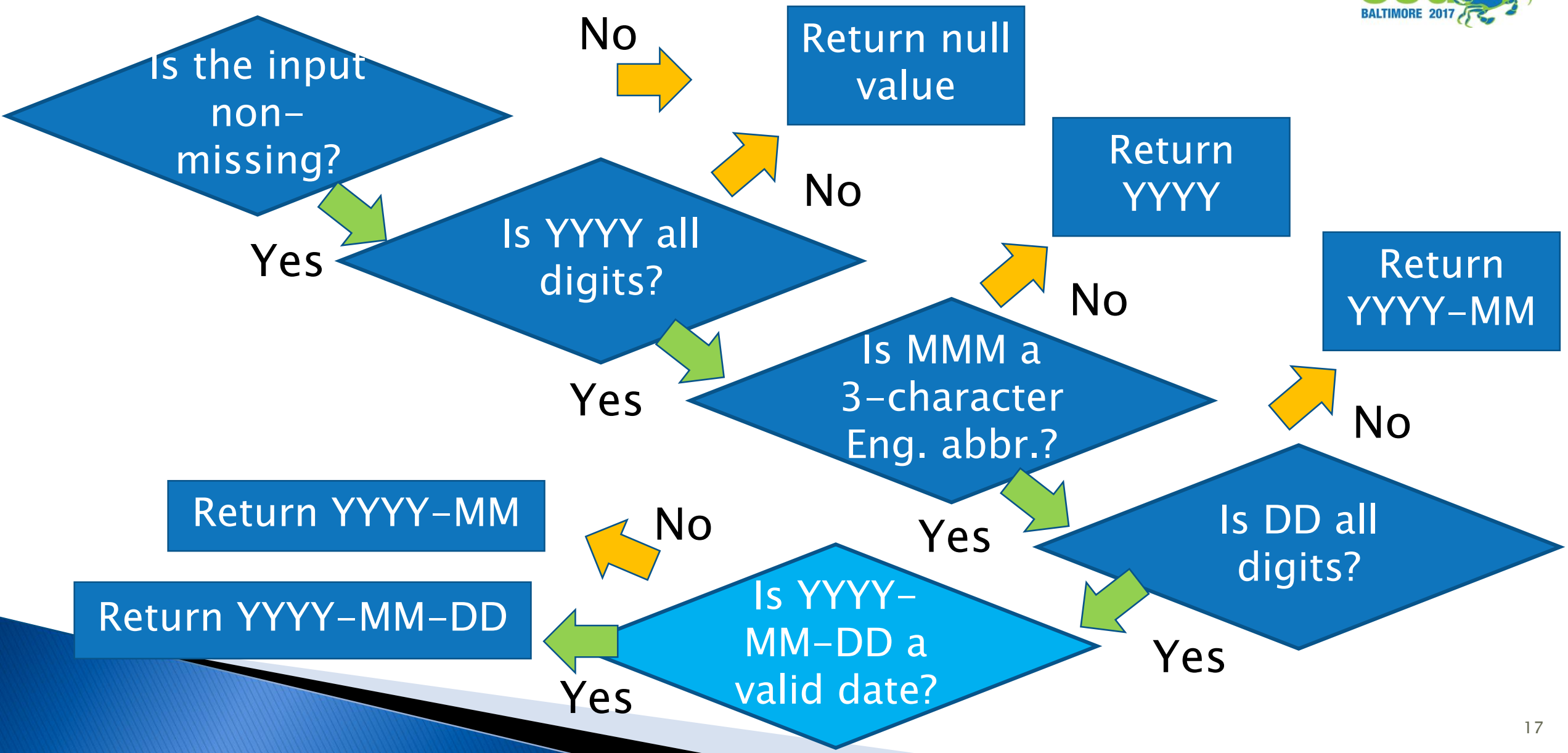
Paper Example: Output

Input	Output
14MAY2017	2017-05-14
14May2017	2017-05-14
UNMAY2017	2017-05
UNUNK2017	2017
14UNK2017	2017
14MAYUKUK	*null*
UNUNKUKUK	*null*
99JAN2017	2017-01-99
31FEB2017	2017-02-31

Matches target output

Invalid dates

Modified Example: Pseudocode



Modified Example: Layer 3

```
outdate = yyyy || '-' || strip(mm) || '-' || dd;
year  = input(yyyy, 8.);
month = input(mm, 8.);
day   = input(dd, 8.);

month_start    = mdy(mo, 1, yr);
month_end      = intnx('month', mo_start, 0, 'end');
month_lastday = day(month_end);

if (day < 1) or (day > month_lastday)
 then outdate = yyyy || '-' || strip(mm);
```

Return YYYY-MM-DD

Is YYYY-MM-DD a valid date?

Return YYYY-MM

Modified Example: Output

Input	Output
14MAY2017	2017-05-14
14May2017	2017-05-14
UNMAY2017	2017-05
UNUNK2017	2017
14UNK2017	2017
14MAYUKUK	*null*
UNUNKUKUK	*null*
99JAN2017	2017-01
31FEB2017	2017-02

All dates are valid

(Could still use a data query: "2017-02-28"?)

Summary

- PROC FCMP
  - Program custom functions to use *within* a DATA or PROC step
  - Increase modularity and reusability of code
  - Syntax similar to DATA step syntax
- Partial dates in SDTM data sets
  - ISO8601 format (e.g. 2017-05)
  - Imputation not allowed by FDA
- Converting partial dates
  - Need to anticipate missing values and missing value codes
  - Avoid outputting invalid dates (e.g. 2017-01-99)

Name: Noory Kim
Organization: CROS NT LLC
Address: 501 Eastowne Drive
City, State ZIP: Chapel Hill, NC 27514
Work Phone: 919-929-5015
E-mail: noory.kim@crosnt.com
Web: www.crosnt.com