



PHUSE CT01: SAS Formats on the Fly using ADaM Datasets

John Hendrickx
Senior Statistical Programmer
Danone Nutricia Research
Utrecht, The Netherlands

SAS User-defined Formats

- Assign a label to a value
- Example:

```
proc format;  
  value trtnF  
    1='Active'  
    2='Control'  
  ;  
run;
```

- Assign the format in a data step:

```
data ads1;  
  set adam.ads1;  
  format trt01pn trtnF.;  
run;
```

SAS User-defined Formats

- Useful for
 - Smaller datasets
 - Ordering categories
 - Typing shortcut

SAS User-defined Formats

- What's not to like?
 - Formats are in an external file
 - SAS include file
 - Format catalog
 - SAS dataset with CNTLIN option in PROC FORMAT

Alternative

- External file is unnecessary
 - If ADaM conventions are used
 - Information is there
 - TRT01P and TRT01PN
 - AVISIT and AVISITN
 - Extract numeric values and character counterparts
 - Create format on the fly
 - In a macro (of course)

Numeric/Character Pairs

- Use PROC SORT with the NODUPKEY options

```
proc sort data=adam.adsl  
          out=fmtdata (keep=trt01pn trt01p) nodupkey;  
  by trt01pn trt01p;  
run;
```

Numeric/Character Pairs

Obs	trt01pn	trt01p
1	1	PLACEBO
2	2	RA6MG
3	3	RA12MG

Formats on the Fly

- Call execute to turn dataset into format statements

```
data _null_;  
  set fmtdata end=eof;  
  if _N_=1 then call execute("proc format; value trt01pnF");  
  call execute(cats(trt01pn, '=""', trt01p, '"')');  
  if eof then call execute('; run;');  
run;
```

Call Execute Report

- In the log:

NOTE: CALL EXECUTE generated line.

```
1          + proc format;  
1          +                               value trt01pnF  
2          + 1="PLACEBO"  
3          + 2="RA6MG"  
4          + 3="RA12MG"  
5          + ;
```

NOTE: Format TRT01PN has been output.

```
5          + run;
```

As a Macro

```
%macro makeFmt(data=, numvar=, charvar=, fmt=);  
  /* Get unique values of the numeric and character variables */  
  proc sort data=&data out=__fmt(keep=&numvar &charvar) nodupkey;  
    by &numvar &charvar;  
  run;  
  
  /* Use "call execute" to create the format */  
  data _null_;  
    set __fmt end=eof;  
    if _N_=1 then call execute("proc format; value &fmt ");  
    call execute(cats(&numvar, '=', &charvar, ' '));  
    if eof then call execute('; run;');  
  run;  
%mend;
```

- Run using:

```
%makeFmt(data=adam.adsl, numvar=trt01pn, charvar=trt01p, fmt=trt01pn)
```

Bells & Whistles

- Delete the temporary dataset
- Include a “where” option
- Smart defaults based on ADaM conventions
 - “&charvar” is “&numvar” minus “n”
 - “&fmt” is “&numvar” plus “F”

Conclusion

- *Imitation is the sincerest form of flattery*
- Why has SAS never imitated SPSS, BMPD, Stata
- Is it a good idea to store formats separately?
- Formats are now one macro away!