

Formatting and Labeling Variables with SAS Data Set Values.

Amie Bissonett, Mayo Clinic, Rochester, MN

Introduction:

The purpose of this paper is to introduce the use of SAS data set values to format, informat and label variables. PROC FORMAT makes use of output and input control data sets to store and receive variable values and construct formats and informats. For labeling variables, a simple approach involves reading data set values into a macro variable(s) and resolving the values within a LABEL statement.

Formats & Informats:

SAS formats and informats are used to write and read, respectively, variable values in a pre-defined way. Sometimes the values needed to format or informat a variable already exist in a data set, or they can be created as an output data set from a procedure. It is convenient to use the data set to create the format, rather than manually typing it in, especially if it's a repetitive task.

The CNTLOUT= option of PROC FORMAT enables you to produce an output control data set, which contains information about the informats and formats in the FORMATS catalog of the specified library, or the WORK library if no LIBRARY option is specified.

❖ Example 1: Creating an output control data set with CNTLOUT= option

```
proc format cntlout=testcntl;
```

```
value xx  
1='UNO' 2='DOS' 3='TRES'  
other='Can't count that high';
```

```
value $ yy  
'M'='Guy' 'F'='Chic';  
  
invalue inxx (upcase)  
'MALE'=0 'FEMALE'=1;  
  
invalue $inyy  
'one'=1 'two'=2 'three'=3;  
  
run;  
  
proc print data=testcntl;  
title 'Output control data set';  
run;  
** See Figure 1 **;
```

The output control data set consists of variables with global information about each format or informat and specific information about each range and value.

The CNTLIN= option specifies an input control data set in the FORMAT procedure. The input data set must conform the certain specifications. Three variables, all of type character, must be in the input control data set: FMT-NAME is the format or informat name, START is the range starting value, and LABEL is the informatted or formatted value. If the input data set contains picture or character formats, or informats, a TYPE variable value must be included.

In the following example, an input control data set is created from the output data set from PROC RANK. The data set used consists of patient numbers and genetic marker data. Each marker loci has two alleles, noted by <marker>_1 and <marker>_2, so nloci (number of loci) is half of the number of marker variables. The macro creates a format based on the ranked values of allele 1 and 2 combined, with missing values set to zero.

Figure 1.

Output control data set

F M T N A M E	S T A R T	E N D	L A B E L	M I N	M A X	D E F A U L T	L E N G T H	F U Z Z	P R E F I X	M U L T	F I L E	N O T E	T Y P E	S E X	E X C E L L	H L O
XX	1	1	UNO	1	40	21	21	1E-12		0		0	N	N	N	
XX	2	2	DOS	1	40	21	21	1E-12		0		0	N	N	N	
XX	3	3	TRES	1	40	21	21	1E-12		0		0	N	N	N	
XX	**OTHER**	**OTHER**	Can't count that high	1	40	21	21	1E-12		0		0	N	N	N	O
YY	F	F	Chic	1	40	4	4	0		0		0	C	N	N	
YY	M	M	Guy	1	40	4	4	0		0		0	C	N	N	
INXX	FEMALE	FEMALE	1	1	40	6	6	0		0		0	I	N	N	U
INXX	MALE	MALE	0	1	40	6	6	0		0		0	I	N	N	U
INYY	one	one	1	1	40	5	5	0		0		0	J	N	N	
INYY	three	three	3	1	40	5	5	0		0		0	J	N	N	
INYY	two	two	2	1	40	5	5	0		0		0	J	N	N	

❖ Example 2: CNTLIN= option

```
%macro locfreq (dataset, nloci, markers);

proc print data=&dataset;
  title "Original Data Set - &dataset";
run;

/** %put &markers;
d4s414_1 d4s414_2 d4s231_1 d4s231_2
**/

%do i = 1 %to &nloci;

/** Create data set with allele 1 and 2 data
set together
**/

data one (keep=mark);
  set &dataset;
  mark= %scan(&markers,2*&i-1);
run;

data two (keep=mark);
  set &dataset;
  mark= %scan(&markers,2*&i);
run;

data loci;
  set one two;
run;

proc rank data=loci out=data1;
  ranks rank;
  var mark;
run;

proc print data=data1;
  title 'Proc rank output - data1';
run;
```

```
data cntlmark (rename=(mark=start
                        rank=label));

  set data1;
  if rank=. then rank=0;
  fmtname="mark&i._";
run;

proc print data=cntlmark;
  title 'Data cntlmark, input control data set';
run;

/** Call to proc format to create formats
from input control data set
**/

proc format cntlin=cntlmark;
run;

%end;    %* %do i = 1 %to &nloci **;

%macro fmts;
  format
    %do j = 1 %to &nloci;
      %scan(&markers,2*&j-1)
      %scan(&markers,2*&j) mark&j._
    %end;
  ;
%mend;

data &dataset;
  set &dataset;
  %fmts;
run;

proc print data=&dataset;
  title "&dataset with formatted values";
run;

%mend locfreq;
```

```
%let marks = d4s414_1 d4s414_2
d4s231_1 d4s231_2;
```

```
%locfreq (Genedat, 2, &marks)
```

Original Data Set - Genedat				
pt	d4s414_1	d4s414_2	d4s231_1	d4s231_2
1	234	235	163	163
2	235	236	163	163
3	234	235	157	165
29				
30				
39	230	233	161	163
43	235	236	163	163

```
** Data for second marker, d4s231 **;
```

Proc rank output - data1		
OBS	MARK	RANK
1		
2	157	1
3	161	2
4	163	3
5	165	4
6	169	5

Data cntlmark, input control data set			
OBS	START	LABEL	FMTNAME
1		0	mark2_
2	157	1	mark2_
3	161	2	mark2_
4	163	3	mark2_
5	165	4	mark2_
6	169	5	mark2_

Genedat with formatted values

pt	d4s414_1	d4s414_2	d4s231_1	d4s231_2
1	3	4	3	3
2	4	5	3	3
3	3	4	1	4
29	0	0	0	0
30	0	0	0	0
39	1	2	2	3
43	4	5	3	3

Labels:

The LABEL statement associates descriptive labels with variables. A valid label is a character string, up to length 40. This string can be assigned with plain text, as well as with values of macro variables. An entire macro variable can be assigned to a label, or a %SCAN can be used to split a list of values into multiple labels. Values from a data set containing descriptions of variables can be stored in a macro variable(s) and then resolved within the LABEL statement. Be sure to use double quotes to resolve the macro variable.

An example of the macro variable syntax:

```
proc print data=y;
  title 'Original Data y';
run;
```

Original Data y	
OBS	VAR1
1	321
2	987

```
%let label1 = Yippy;
```

```
proc print data=y label;
  label var1="&label1";
  title 'Data y with labels';
run;
```

Data y with labels	
OBS	Yippy
1	321
2	987

This next example uses the same original data set as Example 2, Genedat. A second data set contains a variable, marker, with values being the full name of the markers. Those markers with names longer than six characters (since _1 or _2 are added to each variable name) must be abbreviated to accommodate SAS's eight character limit on variable names, so it is useful to add the full name as a label.

❖ Example 3: Creating a string of labels as a single macro variable and using %scan to loop through the labels

```
%macro label (dataset, labfile, marks);
```

```
  proc print data=&labfile;
    title 'Data set containing label values';
  run;
```

```
%* Returns the number of obs in &nobs *;
```

```
%nobs(dsn=&labfile)
```

```
data _null_;
  set &labfile;
  %do i = 1 %to &nobs;
    if _n_=&i then
```

```
      call symput ("m&i",marker);
  %end;
```

```
run;
%let labels= &m1;
```

```
%do i = 2 %to &nobs;
  %let labels = &labels &&m&i;
%end;
```

```
data &dataset;
  set &dataset;
  label
```

```
    %do i = 1 %to &nobs;
      %scan(&marks,2*&i1)=
        "%scan(&labels,&i)"
      %scan(&marks,2*&i)=
        "%scan(&labels,&i)";
    %end;
```

```
;
run;
```

```
proc print data=&dataset label;
  title "&dataset with labels";
run;
```

```
%mend label;
```

```
%let marks = d4s414_1 d4s414_2
d4s231_1 d4s231_2;
```

```
%label (Genedat, marklabl, &marks)
```

Data set containing label values	
OBS	MARKER
1	D4S414
2	D4S231

Genedat with labels				
Patient Number	D4S414	D4S414	D4S231	D4S231
1	3	4	3	3
2	4	5	3	3
3	3	4	1	4
29	0	0	0	0
30	0	0	0	0
39	1	2	2	3
43	4	5	3	3

References:

SAS Institute, Inc. (1990),
SAS Procedures Guide, Version 6, Third Edition, Cary, NC: SAS Institute, Inc.

SAS Institute, Inc. (1990),
SAS Guide to Macro Processing, Version 6, Second Edition, Cary, NC: SAS Institute, Inc.

~ A special thanks to Jon Kosanke for letting me pick his brain when I got stuck.

Author:

Amie Bissonett
Mayo Clinic
200 First Street SW
Rochester, MN 55905
(507) 284-0063

bissoner@mayo.edu

Any comments or questions are welcome.
Problems or complaints... well, I'll take those too.