

Formats and Labels-Making Your Output More Readable

Judith Boura, William Beaumont Hospital, Royal Oak, Michigan

ABSTRACT:

This course teaches you how to use the LABEL and FORMAT procedures to make your output easier to understand when writing programs using SAS® software. You will learn how to assign labels to variables using a simple LABEL statement. You will also learn how to write your own formats, how to assign variables to either formats you create or standard SAS formats and how to create a permanent library of formats so you don't have to keep retyping common formats.

INTRODUCTION:

When working with a data set, you may have hundreds or even thousands of variables to work with. Using only the 8 character variable names that the SAS system allows can be very confusing. Assigning labels and formats to each variable allows you to better describe what each variable represents while still having numeric codes for the variables to be used in analyses in the data set. Labels and formats help make the output easier to read and understand without having to look up what the codes represent. This is especially useful when showing the output to someone not familiar with the database.

LABELS:

Assigning labels to variables is a quick and easy process that may save much confusion in the long run. Adding labels is a very easy step that you add anywhere in your DATA step. It doesn't matter what type of DATA step you are using (MERGE, INPUT, SET), you can add a LABEL statement. An example follows:

```
Data labels;  
Set one.test;
```

```
Label  
Age='The age at randomization'  
on2cath='Onset to cath lab(min)'  
sex='1=male 2=female'  
hospdays='Days in hospital';
```

You may enter as many LABEL assignments as you need. Each assignment starts with the "variable name" "=" and then up to 40 characters for what you want the label to say in single or double quotes. You need to make sure that there is one semi-colon at the end of the last assignment only. If you are using a CARD

statement to input data, the LABEL statement precedes the CARD statement along with any other new variables being created.

To remove a label, you simply assign it a new label with an empty space between the quotes.

Example:

```
Label Age=' ';
```

You can also use the ATTRIB statement in a DATA step to assign labels. An ATTRIB statement allows you to assign lengths, formats, informats and labels to one or more variables all in one statement.

```
ATTRIB "variable name"  
label='enter label here';
```

FORMAT:

Data is generally not entered as text but as codes; 0=no, 1=yes, 1=male, 2=female, etc.. Trying to remember what the codes represent often can be very confusing. You can format variables so that you can use them coded for analysis but the formats are what are printed in the output. There are 4 types of formats:

- 1) Numeric
- 2) Character
- 3) Date and time
- 4) User defined

There are 67 formats of types 1-3 already written into the SAS system that you can use. These formats are listed in the SAS Language manual pages 67-69. They are described in detail on pages 673 to 713.

The fourth type of format, "user defined", is a format that you create. Some common formats might be:

```
0=No 1=Yes  
0=Female 1=Male  
1=mild 2=moderate 3=severe  
0=false 1=true
```

You can create a format to coincide with any variable you are using. Using PROC FORMAT, you create format names and the format for each number entered for that variable. You start with a PROC FORMAT statement followed by a semi-colon and then each format you create gets a line starting with "value" then the name of the format you are creating. You can use up to 8 characters and they must follow the guidelines

for a valid SAS variable name. The format name for character values must have a dollar sign (\$) as the first character. A space follows the format name and then the assignment of the values of that variable.

Example:

```
proc format;
value noyes 0='no' 1='yes';
value usa 1='USA' 0='Foreign';
value rr 1='Randomiz'
2='registry';
value gend 1='Male' 0='Female';
value rand 1='Stent' 2='PTCA';
value ocl 1='occluded'
2='patent';
value type 1='A' 2='B' 3='C';
```

You may have an unlimited number of value statements for each PROC FORMAT statement. These formats can be created before your DATA step and they can be saved into a permanent library. You first use a LIBNAME statement to assign a directory to your LIBRARY.

Example:

```
libname library
'L:/PAMI/StentPam/SAS';
```

In your PROC FPRMAT statement you would then add "lib=library" before the semicolon.

Example:

```
proc format lib=library;
value ira 1='RCA'2='LCX'3='LAD';
value rtlf 1='right' 2='left';
run;
```

To assign any format to a specific variable, you simply type format "variable name" then the name of the format with a period at the end of it followed by a semi-colon. You can assign many variables to the same format by simply listing them after the format statement followed by the name of the format with a period at the end of it and a semi-colon. You can also do multiple assignments with one format statement. You start with FORMAT, then with all the variables you wanted formatted with the first format, followed by the name of the format with a period at the end of it. On the next line you would list the variables you wanted formatted with the next format followed by the name of the format with a period at the end of it and so on. At the very end, you add 1 semi-colon.

Example:

```
format
randate onsetdt mmdyy8.
cost dollar.2
hxmi hypert noyes.
;
```

These format statements would be included in your DATA step. The formats are permanently assigned to those variables in that data set. If you want to assign temporary formats, you add the same FORMAT statement as a line in a procedure.

Example:

```
proc freq;
tables gender;
format gender gend.;
run;
```

These are just two examples of ways to make your programs and output easier to use. There are VALUE, PICTURE and PUT statements that are also part of the FORMAT procedure that were not explained here. See Chapter 18 of the SAS Procedures Guide for more information on the FORMAT procedure. The SAS Language and Procedures manuals are your best source of how to use these and many other ways to simplify your programs.

References:

SAS Institute Inc., SAS® Language: Reference, Version 6, First Edition, Cary, NC: SAS Institute Inc., 1990. 1042pp.

SAS Institute Inc., SAS® Procedures Guide, Version 6, Third Edition, Cary, NC: SAS Institute Inc., 1990. 705pp.

SAS is a registered trademark of SAS Institute Inc. in the USA and other countries.

Contact:

Judith Boura, Cardiology Research,
William Beaumont Hospital, 3601 W. 13 Mile,
Rd. Royal Oak, MI 48073
(248) 551-0789 jboura@beaumont.edu