SCstatisticalprogramming.com

Store your user-defined formats permanently and Referencing/using Your User-defined Formats

Any time you use PROC FORMAT to create a format, the format is stored in a format catalog. If the SAS data library does not already contain a format catalog, SAS automatically creates a default format catalog named Work.Formats.

As we know, for SAS default library Work, any data stored in it exists temporarily for the current SAS session. Same for formats, any format that is stored in Work.Formats is a temporary format that exists only for the current SAS session. At the end of the current session, the catalog is erased.

If you want to store your user-defined format permanently so they can be used in a new SAS session without re-creating them, you will need to store your formats permanently.

Store your user-defined formats permanently

Here are the steps to do this:

1. Create a library reference (libref, which is the name of the library) to indicate where you want to store your SAS formats using Libname statement, which associates the libref with the permanent SAS data library in which the format catalog is to be stored. This can be the same library where you store your permanent SAS data sets.

```
libname myfmts “c:\SAS learning\formats”;

                 |                      |

        Library Name          Directory
```

(The detailed demonstration on LIBNAME statement is shown in the lecture “how to Creating Permanent SAS datasets using LIBNAME Statement” of my course ‘SAS programming for beginners’. You may go to my website http://www.treehouseofsasprogramming.com/free-sas-tutorials/or my YouTube channel “treehouseofSASprogramming” to watch the free copy of this video)

2. Use the option LIBRARY=libref when you run PROC FORMAT. (Remember, you only need to run this procedure once.)

```
proc format library=myfmts;

        value $genderf   'M' = 'Male'

                         'F' = 'Female‘

run;
```

Referencing/using Your User-defined Formats

Anytime you want to use a SAS data set with associated user-defined formats in a new SAS session, you need to tell SAS where to look for these formats. What you need to do is to assign the same libref again using Libname statement, in this case the library name is Myfmts. Then you can place the FORMAT statement in either a DATA step or a PROC step to associate a format with a variable permanently or temporarily like what I demonstrated in previous tutorials.

SAS code example:

```
libname Myfmts 'c:\SASlearning\formats';


proc print data = score;
        format gender $gender.;
run;
```

SAS searches for the format in two libraries, in this order:

- the temporary library referenced by the libref Work
- a permanent library referenced by the libref Library

That’s why it is recommended, but not required, that you use the word Library as the libref/library name when creating your own permanent formats. I used Myfmts as the library name for demonstration purpose.

Referencing/using Your User-defined Formats (cont.)

If you want SAS to also look in one of your own libraries, you need to issue a FMTSEARCH= option. You can list one or more libraries for SAS to search using this option. For example, if you want to use the formats you placed in the Myfmts library, you would need to submit the following code:

```
options fmtsearch=(myfmts);
```

If you do this, SAS first looks in the Work library, then the library called Library, and then the Myfmts library. If you want SAS to look in the Myfmts library before it looks in either of the other two libraries, you can name them on the FMTSEARCH statement like this:

```
options fmtsearch=(myfmts work library);
```

Now, SAS searches the Myfmts library first and then the Work and the Library libraries.

Displaying format definitions in a user-created library

A useful PROC FORMAT option is FMTLIB. This option creates a listing of each format in the specified library with the ranges and labels. As an example, if you want to display the definitions of all the formats in your Myfmts library, you would submit the following code:

```
title "Format Definitions in the MYFMTS Library";

proc format library=myfmts fmtlib;

run;
```