

PhUSE 2015

Paper CS03

MULTILABEL option – valuable addition to PROC FORMAT (Frequency analysis with multilabel formats)

Anna Amosova

ABSTRACT

PROC FORMAT is a powerful procedure in SAS which allows users to view and summarize data in various ways without creating new variables in the data step. However, until recently this absolutely necessary tool had very significant limitations on capabilities to work with multilabel formats. The addition of the MULTILABEL option has made PROC FORMAT even more powerful by allowing formats to have duplicate and/or overlapping ranges. This paper will give examples how to create multilabel formats, including how to do it from the existing dataset, how to use MULTILABEL option for summarizing data via PROC MEANS, PROC SUMMARY, PROC TABULATE and how to use it with PROC REPORT in SAS 9.4. Finally, this paper will show what errors and unexpected results may appear if MULTILABEL option is used in improper way.

INTRODUCTION

PROC FORMAT is an absolutely necessary tool for the SAS programmer. This procedure had two limitations, until recently, that resulted in extra work. One was that it could accept only mutually exclusive categories. The second was that multiple labels could not be specified for the same value. Multilabel formats allow a SAS user to display over-lapping ranges. Among other things, multilabel formats allow a user to create hierarchical formats, showing multiple levels of groups and sub-groups in the same table. Multilabel formats have been available since SAS Version 8, and are currently compatible with PROC TABULATE, PROC MEANS, PROC SUMMARY and PROC REPORT.

This paper examines the MULTILABEL option for procedures mentioned above and its application to Clinical Trial reporting.

MULTILABEL FORMATS

To describe the main advantages of MULTILABEL format consider the following two things which previously could not be done in the PROC FORMAT:

```
proc format;  
  value ONE  
    0 - 5 = "Under 5"  
    0 - 9 = "Under 9";  
  value TWO  
    1 = "One"  
    1 = "Again One";  
run;
```

(1) Format ONE specifies overlapping ranges

(2) Format TWO assigns different labels to identical values

Both ONE & TWO were not allowed in PROC FORMAT before since it would result in ERRORS if an attempt to create ONE and TWO were made.

However, these errors could be successfully avoided adding MULTILABEL option to the PROC FORMAT. To describe how it works let us create dummy dataset "Demo". Suppose our dataset has 15 patients under 2 treatment groups.

```

proc format ;
  value trtfmt
    1 = "Placebo"
    2 = "Treatment";

  value gender
    1 = "Female"
    2 = "Male";
run;

data demo (label = "Demo" drop=seed1 seed2);
  retain study 100001 seed1 999 seed2 123456;
  do patient = 1 to 15;
    call ranuni(seed1,sex);
    sex = mod(ceil(100*sex),2) + 1;
    trt = mod(patient,2) + 1;
    call ranuni(seed2,age);
    age = ceil(100*age);
    weight = int(ranuni(0) * 30 + 50) + int(ranuni(0) * 9 + 1)*0.1;
    height = int(ranuni(0) * 30 + 150);
    format sex gender. trt trtfmt.;
  output ;
end;
run;

```

The resulting dataset is the following.

	studyid	patient	sex	trt	age	weight	height
1	10001	1	Female	Treatment	74	52.9	163
2	10001	2	Female	Placebo	28	54.2	152
3	10001	3	Male	Treatment	71	75.5	156
4	10001	4	Female	Placebo	32	67.5	171
5	10001	5	Male	Treatment	37	59.3	174
6	10001	6	Male	Placebo	11	66.3	154
7	10001	7	Female	Treatment	4	55.6	176
8	10001	8	Male	Placebo	54	77.7	172
9	10001	9	Male	Treatment	38	66.5	163
10	10001	10	Female	Placebo	5	58.7	158
11	10001	11	Female	Treatment	80	52.4	174
12	10001	12	Male	Placebo	96	67.7	150
13	10001	13	Male	Treatment	65	52.6	158
14	10001	14	Female	Placebo	34	65.4	166
15	10001	15	Male	Treatment	34	73.3	163

To begin with, assume that we need to summarize patients by age groups:

Table 1. Number and Percentage of Patients by Age Group

Age Group	Placebo	Treatment
Children (under 18)	x	x
Adult (18 and above)	x	x
Under 25	x	x
Under 50	x	x
Under 75	x	x
Above 100	x	x

In order to make such a simple table the first thing which comes to mind is the following code:

```

data demo1;
  set phuse.demo;
  length group $20 ;
  if 0 < age < 18 then do;
    group = "Children (under 18)";
    output ;
  end;
  if age >= 18 then do;
    group = "Adult (18 and above)";
    output;
  end ;
  if 18 <= age < 25 then do;
    group = "    Under 25";
    output;
  end;
  if 18 <= age < 50 then do;
    group = "    Under 50";
    output;
  end;
  if 18 <= age < 75 then do;
    group = "    Under 75";
    output;
  end;
  if age >100 then do;
    group = "    Above 100";
    output;
  end;
run;

ods pdf style = sasweb file = "proc freq.pdf";
proc freq data = demo1;
  tables group*trt/ nocol norow nopercent;
run;
ods pdf close;

```

The code above will produce the following output.

Frequency	Table of group by trt			
group	trt			
	Placebo	Treatment	Total	
Under 50	3	3	6	
Under 75	4	6	10	
Adult (18 and above)	5	7	12	
Children (under 18)	2	1	3	
Total	14	17	31	

As we see from the table, the result has a number of drawbacks. First of all the groups "Under 25", "Above 100" are missing because no patients between 18 and 25 or above 100 are present in the dataset Secondly, the groups are not sorted. Finally, categories "Under 50" and "Under 75" are displayed as separate groups, but not as subgroups of "Adult" category.

Below we will show how to deal with all these issues with the main focus of this paper – multilabel formats.

To transform a "simple" format to a multilabel format all we need to do is just add MULTILABEL option to the value statement as shown below:

```

value agefmt (multilabel notsorted)
0 - 17 = "Children (under 18)"
18 - high = "Adult (18 and above)"
18 - 24 = " Under 25"
18 - 49 = " Under 50"
18 - 74 = " Under 75"
101 - high = " Above 100";

```

The NOTSORTED option is an option, which turns off the setting of sorting variable by its formatting values. Later on it will be very helpful. Now, as soon as we have specified the MULTILABEL option we can define overlapping ranges and duplicate values with different labels within a format. Note that if one tried to create a multilabel format without specified option then this would result in an error in the log window that there are over-lapping ranges, and the procedure will not run.

PROC MEANS/SUMMARY WITH MLF* OPTION

Let us first recall that PROC MEANS with a NOPRINT option is identical to PROC SUMMARY procedure. Therefore, all the techniques applied to PROC MEANS throughout this paper are also valid for PROC SUMMARY. Now, consider the following code.

```

proc means data = demo completetypes noprint data = interim;(1)
class age/ mlf preloadfmt; (2)
class trt;
var patient;
output out = interim(drop = _type_ _freq_) n = N;
types age*trt;
format age agefmt.;
run;

```

The most important lines for us are (1) and (2). The first line carries the COMPLETETYPES option. It means that all possible combinations of the CLASS variable values have to be considered. The second line contains MLF and PRELOADFMT options. The former option enables PROC MEANS to use AGE variable with multilabel format and the latter preloads AGEFMT format before analyzing data. Since we are only interested in the counts in each category, the only N Statistic is requested. The TYPES statement requests the crossings between the age categories and TRT. Table 2 presents the result of the INTERIM dataset:

	age	trt	N
1	Above 100	Placebo	0
2	Above 100	Treatment	0
3	Under 25	Placebo	0
4	Under 25	Treatment	0
5	Under 50	Placebo	3
6	Under 50	Treatment	3
7	Under 75	Placebo	4
8	Under 75	Treatment	6
9	Adult (18 and above)	Placebo	5
10	Adult (18 and above)	Treatment	7
11	Children (under 18)	Placebo	2
12	Children (under 18)	Treatment	1

Now it is only left to transpose the N variable and sort the resulting dataset by age.

```

proc transpose data = interim out=interim_tr(drop=_NAME_);
id trt;
by age;
var N;
run;

```

(*) MLF = Multilabel format;

```
proc sql;
  create table result as
  select *
  from sum_tr
  order by input(age,ageinf.);
quit;
```

Above the AGEINF informat is defined as

```
invalue ageinf
"Children (under 18)" = 1
"Adult (18 and above)" = 2
"    Under 25" = 3
"    Under 50" = 4
"    Under 75" = 5
"    Above 100" = 6;
```

The resulting dataset looks as follows:

	age	Placebo	Treatment
1	Children (under 18)	2	1
2	Adult (18 and above)	5	7
3	Under 25	0	0
4	Under 50	3	3
5	Under 75	4	6
6	Above 100	0	0

Notice that the “Under 25” and “Above 100” age categories are added to the dataset even though there was no one in the data set that qualified for those age categories. This is a result of the COMPLETETYPES option, which used PRELOADFMT to pick up this age category from the AGEFMT format. Also note that the number of patients are summarized correctly using the overlapping age ranges specified in the multilabel AGEFMT format.

Further, here we would like to discuss outputs with different ods destinations. If we use LISTING destination then the result will be as we expect. Below you can see listing output and the code, which produces this output.

```
ods listing;
proc print data = result;
run;
ods listing close;
```

Obs	age	Placebo	Treatment
1	Children (under 18)	2	1
2	Adult (18 and above)	5	7
3	Under 25	0	0
4	Under 50	3	3
5	Under 75	4	6
6	Above 100	0	0

However, if we use PDF destination then the indentations for the subgroups will disappear

```
ods pdf style = sasweb file = "proc means.pdf";
proc print data = result;
run;
ods pdf close;
```

Obs	age	Placebo	Treatment
1	Children (under 18)	2	1
2	Adult (18 and above)	5	7
3	Under 25	0	0
4	Under 50	3	3
5	Under 75	4	6
6	Above 100	0	0

So one must be aware of some changes that might happen and must take a precaution before using different ods destinations.

PROC TABULATE WITH MLF OPTION

In this section we would like to consider the use of multilabel formats with PROC TABULATE. The application of MLF option is the same as in PROC MEANS goes together with PRELOADFMLT option.

```
ods pdf style=sasweb file = "proc tabulate.pdf";
proc tabulate data = demo noseps;
  class age/ mlf preloadfmt order = data;
  classlev age/ style = {asis = on};
  class trt;
  table (age=" "),(trt=" ")*n=" " / printmiss misstext="0" box = "Age
Group";
  format age agefmt.;
run;
ods pdf close;
```

Also we would like to stress here that PRELOADFMT and PRINTMISS options are essential if one wants to include categories which do not occur in the dataset. In our case, they are "Under 25" and "Above 100". Further, since we want to sort rows as they go in the definition of our format, namely by data, we include the NOTSORTED option in the VALUE statement. This option maintains the sort order of rows exactly as specified in the format instead of sorting them by formatted values, which is a default ordering when formats are used. In addition, the ORDER=DATA option has to be specified in the CLASS statement of PROC TABULATE to honor the order of our format. Another popular task, which is commonly encountered in Clinical Trials reporting, is to produce tables with subgroups. This can be handled by specifying the CLASS variable in the CLASSLEV statement with ASIS=ON style as shown above. Below you can find the produced output.

Age Group	Placebo	Treatment
Children (under 18)	2	1
Adult (18 and above)	5	7
Above 25	5	7
Above 50	2	4
Above 75	1	1
Above 100	0	0

Like in the previous section, here again we face some problems related to ods destinations. This time if one changed the PDF destination to the LISTING destination then subgroups will disappear even though one runs the program with style, which turns on indentation.

Age Group	Placebo	Treatment
Children (under 18)	2.00	1.00
Adult (18 and above)	5.00	7.00
Under 25	0	0
Under 50	3.00	3.00
Under 75	4.00	6.00
Above 100	0	0

Therefore we would like to stress again that you should be careful when choose ods destinations and a final result depends on it a lot. However, in all the rest of the programs, which are coming in the next sections, there will be no such differences. So we will provide examples with PDF destinations only.

PROC REPORT WITH MLF OPTION

For the moment, we have considered multilabel formatting for PROC MEANS, PROC SUMMARY, and PROC TABULATE, but fortunately starting from SAS 9.3 another very powerful procedure has acquired the capability to work with the “magic” MLF option, that is the PROC REPORT procedure.

In this section, we are going to use the PROC REPORT procedure to count the number of patients for each age group and compute the mean height and weight within the groups. In addition patients will be stratified by gender.

```
ods pdf style = sasweb file = "proc report.pdf";
proc report data = phuse.demo nowd completerows;
    column sex age patient height weight;
    define sex/ group;
    define age/ group mlf preloadfmt format = agefmt. order = data
style(column) = {asis = on} "Age Group";
    define patient/ n "Number of patients";
    define height/ mean format=6.2 "Mean Height (cm)";
    define weight/ mean format=6.2 "Mean Weight (kg)";
    rbreak after/ summarize;
run;
ods pdf close;
```

The key options here are COMPLETEROWS, PRELOADFMT and MLF. The first two options are used together to display all possible combinations of the values of the group variables, even if one or more combinations do not occur in the input dataset. Another important option we use is MLF option. It is exactly this option, which allows us to apply multilabel format AGEFMT and compute frequencies and means for overlapping categories in the right manner. Next, it is worth mentioning here that multilabel formatting is only applied to a group or across variable. Otherwise, if you try to apply the MLF option to any other definition, a warning message will be produced. For instance, if the group variable were changed to an order variable for the DEFINE AGE statement, the SAS log would show the following warning:

WARNING: The MLF option is valid only with GROUP and ACROSS variables. MLF will have no effect for the variable age.

Further, since GROUP option orders rows in a report according to their formatted values, we will have the following group order. “Above 100”, “Under 25”, “Under 50”, “Under 75” go first and then they are followed by “Adult” and “Children” categories. To fix it you add NOTSORTED option to the VALUE statement in the definition of your format and insert ORDER=DATA option to the DEFINE AGE statement.

The produced output is shown below:

sex	Age Group	Number of patients	Mean Height (cm)	Mean Weight (kg)
Female	Children (under 18)	2	167.00	57.15
	Adult (18 and above)	5	165.20	58.48
	Under 25	0	.	.
	Under 50	3	163.00	62.37
	Under 75	4	163.00	60.00
	Above 100	0	.	.
Male	Children (under 18)	1	154.00	66.30
	Adult (18 and above)	7	162.29	67.51
	Under 25	0	.	.
	Under 50	3	166.67	66.37
	Under 75	6	164.33	67.48
	Above 100	0	.	.
		15	163.33	63.04

Finally, if you need to create a detailed report instead of a summary report, you can change any other group variable to an order variable or add an order variable itself.

CNTLIN DATASETS AND MULTILABEL FORMATS

Sometimes it is needed to create a format from an existing dataset to avoid unnecessary programming. Here is an example how to create formats from datasets using CNTLIN= option in the PROC FORMAT statement. These datasets are called 'input control' datasets, and each one contains one or more formats. Consider the following example:

```
data format;
  fmtname = "fmt";
  length label $20. hlo $3.;
  start = 0;   end = 17; hlo = "SM";   label = "Children (under 18)";
  output;
  start = 18;  end = .;  hlo = "SMH"; label = "Adult (18 and above)";
  output;
  start = 18;  end = 24; hlo = "SM";   label = "    Under 25";
  output;
  start = 18;  end = 49; hlo = "SM";   label = "    Under 50";
  output;
  start = 18;  end = 74; hlo = "SM";   label = "    Under 75";
  output;
  start = 101; end = .;  hlo = "SMH"; label = "    Above 100";
  output;
run;
```

While START, END and LABEL variables are standard ones for creating input control datasets, let us focus on the main attribute of this dataset, namely HLO= option. HLO = "S" specifies that NOTSORTED option is in effect, HLO = "M" declares that we are going to create a multilabel format and HLO = "H" says that upper limit for a range equals high. Next, to create a control input dataset using CNTLIN= option you proceed as follows:

```
proc format lib = work.formats cntlin=format;
run;
```

It is easy to check that FMT and AGEFMT formats are identical which is left to the reader.

CONCLUSION

In conclusion we would like to say that the MULTILABEL option is a useful addition to the FORMAT procedure. When used with the MLF enabled procedures it has useful applications in clinical trials reporting. Also it enables to program in more efficient and low cost way without creating extra variables which saves user's time and efforts. Lastly, we do hope that in the future it will be extended to other procedures such as PROC FREQ and others.

REFERENCES

G. D. Babcock, Multiple Facts about Multilabel Formats, New Your State Department of Health, Troy, NY, NESUG 2008
R. Cody, Learning SAS by Example: A Programmer's Guide, Cary, NC: SAS Institute Inc., 2007
T. Chapman, Making the Most Out of Multilabel Formats, Oregon Department of Consumer and Business Services, Salem, OR.

ACKNOWLEDGMENTS

Special thanks to Peter Lord and Igor Savostianov for their comments and suggestions while preparing this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Anna Amosova
Experis Clinical
43/2 Gagarin Avenue Kharkiv 61001, Ukraine
Work Phone: +380577552020 ext.2404
Email:anna.amosova@intego-group.com

Brand and product names are trademarks of their respective companies.