Applying INVALUE STATEMENT in PROC FORMAT for Ordering Character Variables

Prajitha Nair, Kreara Solutions, Thiruvanthapuram, India

ABSTRACT

PROC FORMAT is one of the most useful and powerful SAS® tools. The FORMAT procedure enables the SAS programmers to define their user-defined formats and informats for variables to assign the data labels to the data values used during programming. This procedure acts as a catalyst for handling different situations. While generating the report for the TFLs in clinical trial projects using PROC REPORT, the programmers especially those who are novice in SAS consume most of the time in ordering the values in the desired way,. This paper explains the creation of ordering values for character variables using INVALUE statement of PROC FORMAT, rather than using IF-THEN/ELSE or SELECT statement. The ordering of character variables in a manner different from alphabetical order through the application of user defined informats and formats is described. Thus, the areas encompassed in this paper are as follows:

- Creation of user-defined informats and formats using FORMAT procedure
- Usage of INPUT and PUT functions
- Creation of ordering variables (numeric type) using assignment statement and above mentioned functions in a DATA step
- Generation of report using the variables thus created in REPORT procedure
- Overview of options related to PROC FORMAT

The above mentioned topics are well-known to all the SAS programmers but this paper will be focusing on how all these can be used effectively to present the values of the variables in certain order while programming for tables and listings.

INTRODUCTION

While generating the tables and listings using PROC REPORT in reference to the layout, the layout suggests that the values in the column for statistics, treatment groups, vital signs or ECG parameters, visits etc are to be presented in a desired order. The programmers always spend much of the time in manipulation of data to obtain the desired ordering structure. The programmers are more inclined towards using the IF-THEN/ELSE statement due to the ease of use. However, this approach cannot be considered as a good programming practice. The FORMAT procedure is an alternative approach to handle the situation in a more optimized manner. We shall go through examples to illustrate the importance of INVALUE statement. Unlike alphabetical order, a variable containing the sequence corresponding to the data values of character type will be created.

Consider the following three scenarios wherein the converted values of a variable are to be stored into a new variable.

- Convert the values M and F as Male and Female
- Convert the values 1 and 2 as Male and Female
- Convert the values M and F as 1 and 2

In the first two cases, we can easily convert character variables or numeric variables into character variables using VALUE statement in PROC FORMAT, but we can observe that in the third case the conversion of character variables to numeric variables cannot be accomplished using the VALUE statement. The programmer uses the IF-THEN/ELSE statement or SELECT statement in such scenarios. This paper illustrates the use of INVALUE statement in PROC FORMAT to accomplish the output in the desired way.

EXAMPLE

The following layout presents the summary statistics for Vital Signs parameters. It is required that the statistics be presented in the same order mentioned in the layout.

Sponsor-name Draft
Protocol ID : XXXXXXXX Page X of Y

Table 14.3.5.3 Summary of Vital Sign parameters by regimen and time-point (Safety Population)

Parameter	Time Point	Statistics	Trt A (N=XX)	Trt B (N=XX)	Trt C (N=XX)	Trt D (N=XX)	Trt E (N=XX)	Active (N=XX)	Placebo (N=XX)
HR (bpm)	Screening	N	XX	XX	XX	XX	XX	XX	XX
		Mean	XX.X	XX.X	XX.X	XX.X	XX.X	XX.X	XX.X
		SD	XX.X	XX.X	XX.X	XX.X	XX.X	XX.X	XX.X
		Minimum	XX.X	XX.X	XX.X	XX.X	XX.X	XX.X	XX.X
		Median	XX.X	XX.X	XX.X	XX.X	XX.X	XX.X	XX.X
		Maximum	XX.X	XX.X	XX.X	XX.X	XX.X	XX.X	XX.X

Note to the Programmer:

1. Please include all the scheduled Time Points
2. Please present the statistics in the same order: N , Mean, SD, Minimum, Median and Maximum
3. Please present all the vital sign parameters that have been assessed in this study.

Note: HR = heart rate, SBP = systolic blood pressure, DBP = diastolic blood pressure, Resp .= respiration rate.

DDMMMYYYY:HH:MM

Applying appropriate DATA step manipulations and appropriate procedures can help the programmer to obtain the desired results. Further, the REPORT procedure can be used to present the report as per layout. The programmer should be well-versed with the usage of ORDER variables and ORDER= option in the DEFINE statement to ensure that the results are reported in meaningful order. In some cases however the use of ORDER variables and the ORDER= option may not give the desired results. The programmers tend to attain these results by generating an additional numeric sequence variable through DATA step manipulation. The programmers usually resort to using the IF/THEN ELSE statement or SELECT statement to create the ordered variable of the numeric type.

ALTERNATIVE APPROACH

There is another alternative method to convert the character variables to numeric and storing them as ordering variables and then mapping values to these order variables. The INVALUE and VALUE statement in the FORMAT procedure enables to accomplish these and produce the desired report without much effort. This paper emphasizes on the creation of ordered variables of numeric type and subsequently uses them in the PROC REPORT. This paper illustrates the steps involved to create the numeric variables to present the column ‘STATISTICS’ in a desired format with respect to the layout.

FIRST STEP

The summary statistics is obtained using appropriate procedures like PROC MEANS or PROC SUMMARY at first. While presenting the summary statistics, standard rules regarding the decimal point needs to be considered. Standard rules states that:

- Mean and Median will be presented with one more than the number of decimal values in the observed variable
- SD will be presented with two more than the number of decimal values in the observed variable

- Minimum and Maximum will be presented with same number of decimal places as in observed value

The output datasets generated when the procedures are executed will contain all the summary statistics as different variables along with other relevant variables. As part of the reporting requirement, we will perform a transpose on the output dataset in order to obtain a vertical data structure with a new variable _name_ being created containing labels for the statistics. In this particular example, the labels for the statistics are given as countc for N, meanc for mean, sdc for standard deviation and so on.

The snippet of code which calculates the summary statistics followed by the decimal alignment is as follows:

```
PROC MEANS DATA=vital NOPRINT;
  BY d_parmc visitc;
  VAR d_value;
  OUTPUT out=vital_desc n=count
                        mean=avg
                        std=sd
                        median=median
                        min=mini
                        max=maxi;
RUN;

DATA vital_desc;
  SET vital_desc;

  countc = put(count,best.);
   meanc = put(avg,5.1);
     sdc = put(sd,5.2);
  median = put(median,5.1);
    minc = put(mini,5.1);
    maxc = put(maxi,5.1);

  DROP count avg median sd mini maxi;
RUN;

PROC TRANSPOSE DATA=vital_desc OUT=vital_desctrn;
  BY d_parmc visitc;
  VAR countc meanc sdc medianc minc maxc;
RUN;
```

In the transposed dataset the name of the variable is stored as the values in the variable called _NAME_ and the values of these will be in the variables named as COL1. But we can see that the sequence of the values in _NAME_ will not be in the desired format. We need to present the variable _NAME_ as STATISTICS (third column in the layout) in the order as shown in the layout above. The steps to achieve the desired format are described as below.

CREATION OF USER-DEFINED INFORMATS AND FORMATS USING FORMAT PROCEDURE FOR THE VARIABLE STATISTICS

FORMAT procedure enables to convert the raw data values to the actual or formatted values for the variables. Informats instruct SAS how to read the data and store whereas formats instruct SAS how to display the data values in the SAS dataset. In this paper, the informats aid to convert and the format displays the data values.

Now, we shall understand what format and informat does and how these will be used to implement in this paper.

With formats, one can

- Display the numeric values in character format (for example, 1 as Male and 2 as Female)
- Display the character value in character format (for example, M as Male and F as Female)
- Display the numeric values as numeric value in a particular template (for example, 1 will be printed as 001, 100 as 100% etc)

With informats, one can

- Convert a number to a character string (for example, 1 to Male, 2 to Female)
- Convert a character string to a different character string (for example, 'M' to 'Male')
- Convert a character string to a number (for example, Male to 1)
- Convert a number to another number (for example, temperature ranging from 35.6 to 37.7 (both inclusive) will be flagged as 2, less than 35.6 to 1 and greater than 37.7 as 3)

We can observe that the format is defined to be of character or numeric type based on the data values in the variable on which format is to be applied and the informat is based on the data type that the data values are supposed to be converted. The conversion of one form to another can be done using INPUT and PUT function which will be discussed subsequently in the next section.

The general syntax of FORMAT procedure with selective statement is:

PROC FORMAT <option(s)>;
VALUE <$>*name* <(*format-option(s)*)> *value-range-set(s)*;
INVALUE <$>*name* <(*informat-option(s)*)> *value-range-set(s)*;
RUN;

The programmers are well-versed with VALUE statement of FORMAT procedure. VALUE statement is a tool to create formats that are used to enhance the display of the variable values whereas INVALUE statement is used to define the informats for reading and converting the raw data values into understandable terms.

INVALUE AND VALUE STATEMENT

An informat is created to read and convert the raw data values using the INVALUE statement. In the transposed dataset, the data values for _NAME_ are given as countc for N, meanc for mean, sdc for standard deviation and so on. The values in the _NAME_ column need to be presented in the STATISTICS column in the layout and also in the desired order. To accomplish this, we create a sequence for the values in the variable _NAME_ in line with the layout requirement by creating an informat using the INVALUE statement. An associated format is created for the sequence values using the VALUE statement such that the labels are as per the layout.

```
PROC FORMAT;
  INVALUE fsumm "countc"   = 1
                "meanc"    = 2
                "sdc"      = 3
                "minc"     = 4
                "medianc"  = 5
                "maxc"     = 6
      ;
RUN;
```

Thus, the sequence has been defined using INVALUE statement and we need to create user-defined formats corresponding to these values to map the label to the data values. It is to be noted that a numeric informat is created as the converted values 1, 2, 3 and so on are of numeric type.

```
PROC FORMAT;
  VALUE fsumm 1 = "N"
              2 = "Mean"
              3 = "SD"
              4 = "Min"
              5 = "Median"
              6 = "Max"
   ;
RUN;
```

One-to-one mapping is performed here and the data labels are defined corresponding to each data value. Here we have named both the informat and format as “fsumm”. Now we have to use these user-defined informats and formats for conversion of the data values and stored.

APPLYING THE INFORMAT AND FORMAT TO CREATE THE SEQUENCE VARIABLE

We will now create a numeric variable corresponding to the _NAME_ variable (which has the values countc, meanc, sdc etc.) which can be used as the ordering variable to obtain results in the required sequence. This numeric variable will contain integer values and thus we will now have to apply the format so that appropriate labels are presented in the report. The above are accomplished by the use of the INPUT statement and PUT statement respectively.

The below step shows that a new variable, SUBSEQ, is created corresponding to the _NAME_ variable using the INPUT statement and the informat fsumm. The variable SUBSEQ is a numeric variable which assigns a sequence to the labels in _NAME_.

SUBSEQ = input(_name_,fsumm.);

This step will create a new variable SUBSEQ which will have the following values corresponding to the values in _NAME_.

NAME	SUBSEQ
countc	1
meanc	2
sdc	3
Minc	4
Medianc	5
maxc	6

Thus, in this step, character variable has been converted to numeric variable using the informat.

The next statement helps in providing appropriate labels to the sequence variable, SUBSEQ, as per the layout with the use of the format.

SUBCHAR = put(subseq,fsumm.);

GENERATE REPORT

The newly created variables can then be used in the PROC REPORT to generate the outputs as per the layout. The snapshot of the REPORT procedure showing the corresponding statements and the output generated has been shown below.

```
proc report data=final nowd spacing=1 split='`' headline headskip missing ;
    column seq pagenum schar visitnum visit subseq subchar _1 _2 _3 _4 _5 _99 _6;
    define pagenum /order noprint ;
    define seq /order order=data noprint ;
    define schar /order order=data width=15 "Parameter` ` ` ` " left flow
    spacing=0;

    define visitnum /order order=internal noprint ;
    define visit /order order=data width=10 "Time Point` ` ` ` " left flow ;

    define subseq/order order=data noprint ;
    define subchar /display width=10 left flow "Statistics` ` ` ` " order=data;

    define _1/display "Trt A`(N = &trtcat1)" left flow width=10;
    define _2/display "Trt B`(N = &trtcat2)" left flow width=10;
    define _3/display "Trt C`(N = &trtcat3)" left flow width=10;
    define  4/display "Trt D`(N = &trtcat4)" left flow width=10;
    define  5/display "Trt E`(N = &trtcat5)" left flow width=10;
    define _99/display "Active`(N = &trtcat7)" left flow width=10;
    define _6/display "Placebo`(N = &trtcat6)" left flow width=10;

    break after visitnum/skip;
    break after pagenum /page ;
run;
```

In the above program, the newly created variable SUBSEQ and SUBCHAR are created in the SAS dataset “final” and these variables have been used in the PROC REPORT. The sequence of the variable is not to be printed in the report and thus the NOPRINT option is given for the variable “SUBSEQ”. The values under the column statistic are printed using the variable SUBCHAR.

The output is as follows:

```
Sponsor-name                                                                    Draft
Protocol ID : XXXXXXXX                                                  Page 1 of 25

             Table 14.3.5.4 Summary of Vital Signs by Regimen and Time Point (Safety Population)

Parameter   Time     Statistics Trt A     Trt B     Trt C     Trt D     Trt E     Active    Placebo
            Point              (N = 7)   (N = 6)   (N = 6)   (N = 6)   (N = 6)   (N = 31)  (N = 6)
_____________________________________________________________________________________________________

HR (bpm)   Screening  N          7         6         6         6         6         31        6
                      Mean      65.7      65.7      64.8      72.2      71.7      67.9      71.0
                      SD         9.93     13.57     10.85      9.47     14.12     11.35     16.85
                      Min       54        50        45        57        57        45        54
                      Median    65.0      62.5      69.0      73.5      68.5      66.0      68.0
                      Max       83        91        74        85        92        92       101

           Day -1     N          7         6         6         6         6         31        6
                      Mean      73.6      78.8      74.3      80.0      80.0      77.2      71.8
                      SD        12.16     15.59     15.81      8.67     18.12     13.69     10.50
                      Min       59        66        56        70        61        56        53
                      Median    72.0      75.5      76.0      78.0      76.0      76.0      74.5
                      Max       96       109        92        94       112       112       80

      Note: HR= heart rate, SBP= systolic blood pressure, DBP= diastolic blood pressure, Resp.=
respiration rate.
_____________________________________________________________________________________________________

11JUN2011:12:32
```

OPTIONS

STORING THE FORMAT PERMANENTLY

The formats and informats must be stored permanently as these formats can be used later within the same program or related tables or listings or by different users. This benefit is that we do not need to create the format at the usage point of time. The LIBNAME statement can be used assigning the library reference to the physical storage location of the computer. The general form is

LIBNAME *library-reference* "*path of the file where it is to be stored*"

PROC FORMAT library= library-reference.<catalog>**;**
RUN;

The catalog name is optional. If we specify only library-reference, then a catalog name of FORMATS is created by default.

WHAT WOULD HAPPEN IF THE VALUES IN THE INFORMATS ARE A MIX OF LOWER CASE AND UPPER CASE?

If suppose, we have given the name of the derived variables in mixed case (lower case and upper case) while programming, then we can use UPCASE string function in the INVALUE statement to ensure that the informat is defined appropriately..

```
PROC FORMAT;
  INVALUE fsumm (UPCASE) "COUNTC"  = 1
                         "MEANC"   = 2
                         "SDC"     = 3
                         "MINC"    = 4
                         "MEDIANC" = 5
                         "MAXC"    = 6
      ;
RUN;
```

The UPCASE option first converts the values to upper case before going to check for the range of values. For example, if the _name_ contains the value "Meanc" or "meanc" then the first step would be to convert the value to "MEANC" and then check against the range of values and assign 2 in the dataset.

CONCLUSION

The alternative method of using PROC FORMAT in place of IF-THEN/ELSE or SELECT statement for ordering a character variable in a sequence other than the alphabetical order is much more useful. The suggested method uses user defined informats and formats created using PROC FORMAT.

The approach suggested has various advantages when compared to the IF-THEN/ELSE statement approach. Some of them are listed as follows:

- This consumes less CPU time most often and is less error-prone as it involves simpler programming steps.
- User-defined formats and informats make the program more efficient and enhance the readability of the program in comparison to the other method.
- It also helps improving the quality of the program corresponding to PROC REPORT.
- Another benefit is that we can save the format in the permanent library and can re-use the format within the programs and also in the related tables or listings.

REFERENCES

SAS Institute Inc., SAS OnLineDoc, Version 8, Cary, NC: SAS Institute Inc., 1999.

ACKNOWLEDGMENTS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

I extend my thanks wholeheartedly to all my colleagues who provided support and guidance to complete this paper.

CONTACT INFORMATION

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

Author Name	:	Prajitha Nair
Company	:	Kreara Solutions Pvt. Ltd.
Address	:	T-4, 7th Floor, Thejaswani Building, Technopark, Thiruvanthapuram,
City, State,	:	Thiruvanthapuram, Kerala, India. 695581
Work Phone	:	+91-471-2527640 (Ext: 207)
Fax	:	+91-471-3043049
Email	:	prajitha@kreara.com
Web	:	www.kreara.com

****