Great Expectations of PROC FORMAT
—Applications All You Can Do with Format in SAS System

John He, Octagon Research, Wayne, PA  
Ram Valluru, Octagon Research, Wayne, PA

ABSTRACT

The PROC FORMAT in SAS is something that everybody knows but most people know obliviously. With simple VALUE/INVALUE, PICTURE, or MULTILABEL statement, you can build any user-defined format catalog to help develop a variety of applications you wanted. Of course, there are many other ways to achieve your goals, as you are surely a nimble programmer armed with powerful SAS system. However, PROC FORMAT can offer you additional convenience and robustness not only when you are developing but also maintaining the applications.

This paper will not show you how to create formats/informats or format catalog since it is assuming that PROC FORMAT was every SAS programmer’s first lesson. What I want to share with in this paper is to rediscover the Great Expectations of PROC FORMAT—the many potentials that PROC FORMAT can do for you—these Great Expectations might have been just twinkling along with other shining features of SAS system. The great potentials of PROC FORMAT that will be introduced in this paper include the use of formats to generate titles, to create summary statistics with overlapped levels, and to standardize the clinical data.

INTRODUCTION

The SAS system developed its formats and informats for reading and writing data in predefined standard structure. The PROC FORMAT in SAS is something that everybody knows but most people know obliviously.

There are many more and richer user-defined formats have been developed as SAS applications go prospered in many areas. With simple VALUE/INVALUE, PICTURE, or MULTILABEL statement, you can build any user-defined format catalog to help develop a variety of applications you wanted.

The three-element procedure, namely as Proc Format, Value/Invalue and Run, can give you not only the way to determine the value and type of the data, as its traditional image of “format”, but also the power to develop programming applications for dictionary coding, laboratory data conversion, defining PCSA, and etc, in clinical studies.

Of course, there are many other ways to achieve your goals for you are surely a nimble programmer armed with powerful SAS system. However, PROC FORMAT can offer you additional convenience and robustness not only when you are developing but also maintaining the applications.

USING FORMAT TO GENERATE TITLES AND FOOTNOTES

It is very convenient and useful to create and maintain the titles and footnotes in excel files. Suppose you have the titles stored in an excel file as follows:
You can create a SAS dataset from the titles excel file using the codes below.

```sas
data work.titles;
  INFILE "C:\SAS\Data\Xls\titles.csv" MISSOVER DSD DLM=',' firstobs=2;
  length TYPE PROGNAME OUTPUT TITLE1 TITLE2 TITLE3 TITLE4 TITLE5 TITLE6 TITLE7 TITLE8 $ 200;
  input TYPE $ PROGNAME $ OUTPUT $ TITLE1 $ TITLE2 $ TITLE3 $ TITLE4 $ TITLE5 $ TITLE6 $ TITLE7 $ TITLE8 $;
run;
```

```sas
data fmds;
  set titles;
  FMTNAME='TIT_A';
  TYPE='C';
  START=scan(trim(left(progname)), 2, ' ');
  END=START;
  LABEL=title1;
  LENGTH=80;
  keep FMTNAME START END LABEL TYPE LENGTH output;
  FMTNAME='TIT_B';
  TYPE='C';
  START=scan(trim(left(progname)), 2, ' ');
  END=START;
  LABEL=title2;
  LENGTH=80;
  keep FMTNAME START END LABEL TYPE LENGTH output;
  FMTNAME='TIT_C';
  TYPE='C';
  START=scan(trim(left(progname)), 2, ' ');
  END=START;
  LABEL=title3;
  LENGTH=80;
  keep FMTNAME START END LABEL TYPE LENGTH output;
```

...
Then create a format catalog and store it in the library.

```sas
proc format cntlin = fmds library=test;
run;
```

Before you generate the tables, use the format in the library to create the title/or footnote macro variables.

```sas
call symput('tit1', put(tabno, $tit_a.));
call symput('tit2', trim(left(put(tabno, $tit_b.))));
call symput('tit3', put(tabno, $tit_c.));
call symput('fot1', put(tabno, $foot_a.));
call symput('fot2', trim(left(put(tabno, $foot_b.))));
```

When you generate the tables, use the macro variables for the titles/or footnotes.

```sas
title1 "&tit1";
title2 "&tit2";
title3 "&tit3";
footnote1 "&fot1";
footnote2 "&fot2";
```

Hopefully, you will have the tables as your desired:

<table>
<thead>
<tr>
<th>Preferred term by System organ class</th>
<th>Treatment A</th>
<th>Treatment B</th>
<th>Treatment C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total subjects exposed</td>
<td>445 (100%)</td>
<td>432 (100%)</td>
<td>476 (100%)</td>
</tr>
<tr>
<td>Total subjects with events</td>
<td>23 (5%)</td>
<td>18 (4%)</td>
<td>16 (3%)</td>
</tr>
<tr>
<td>Gastrointestinal disorders</td>
<td>10 (2%)</td>
<td>7 (2%)</td>
<td>8 (2%)</td>
</tr>
<tr>
<td>Inguinal hernia</td>
<td>5 (1%)</td>
<td>4 (1%)</td>
<td>4 (1%)</td>
</tr>
<tr>
<td>Musculoskeletal and connective tissue disorders</td>
<td>11 (2%)</td>
<td>13 (4%)</td>
<td>12 (3%)</td>
</tr>
<tr>
<td>Intervertebral disc protrusion</td>
<td>4 (1%)</td>
<td>6 (2%)</td>
<td>7 (2%)</td>
</tr>
</tbody>
</table>

A subject who reported two or more AEs in the same system organ class/term was counted only once in that class/term.

**USING FORMAT TO CREATE SUMMARY STATISTICS**

You can use multi-label format to create summary statistics with overlapped levels. The veteran SAS users might have noticed that the differences of the format procedure in SAS version 6.12 and the additional option of MULTILABEL in SAS version 8.2.

Let’s first take a look at the following examples of Version 6.12:

For SAS version 6.12 the range of values must be mutually exclusive, that means that the value cannot have more than one label.
With version 6.12, you may need 4 steps to obtain the summary statistics from the data.

Step 1: Generate a format with unique Treatments and Total.

```sas
Proc format;
  value trtf
    1="Treatment A"
    2="Treatment B"
    3="Treatment C"
    4="Total"
;
run;
```

Step 2: Set data with Treatments and set data again as Total

```sas
data temp;
  set derived.patstat(where=(safety=1));
  output;
  treat=4;
  output;
run;
```

Step 3: Apply format to data

```sas
data temp1;
```

Consider the data like the following:

```
Obs  id   trt  sex  age  inc  score1  score2  score3
1    1    1    F    35   17   7       2       2
2    17   2    M    50   14   5       3       3
3    33   1    F    45   6    7       2       2
4    49   3    M    24   14   7       5       7
5    65   5    F    52   9    4       7       7
6    81   3    M    44   11   7       7       7
7    2    2    F    34   17   6       5       3
8    18   2    M    40   14   7       5       2
9    34   1    F    47   6    6       5       6
10   50   3    M    35   17   5       7       5
11   11   1    F    65   17   7       2       2
12   27   2    M    60   14   5       5       3
13   33   1    F    55   6    7       2       7
14   28   3    M    74   14   7       5       7
15   35   2    F    62   9    4       7       7
16   21   1    F    54   11   7       7       7
17   7    3    F    38   17   6       5       3
18   28   2    M    20   14   7       5       2
19   24   1    F    27   6    6       5       6
20   20   3    M    25   17   5       7       5
```

With version 6.12, you may need 4 steps to obtain the summary statistics from the data.
set temp;  
    trtment=put(treat, trtf.);  
run;

Step 4: calculate statistics

```
proc means data=temp1;  
    class trtment;  
    output out=tot1(where=(trtment^=' ')) n=n;  
run;  
proc print data=tot1;  
run;
```

Output will be like this:

```
Obs    trtment        _TYPE_    _FREQ_     n  
1     Total             1        20      20  
2     Treatment A       1         6       6  
3     Treatment B       1         8       8  
4     Treatment C       1         6       6  
```

Now with SAS version 8.2 or newer, you can take one step to achieve above results.

The following format can be developed using MULTILABEL Format.

```
Proc format;  
    value trtm (MULTILABEL)  
        1="Treatment A"  
        2="Treatment B"  
        3="Treatment C"  
        1-2="Total1"  
        2-3="Total2"  
        1-3="GrandTotal"  
        Other="MissingTreatment"  
    ;  
    run;
```

```
proc means data=test completetypes; class trt/mlf preloadfmt;  
    output out=tot2 n=n;  
    types trt;  
    format trt trtm.;  
run;  
proc print data=tot2;  
run;
```

There are two notes for the preceded program codes. 1) COMPLETETYPES is an option in the Procedures that creates all possible combinations of class variable values. 2) PRELOADFMT is an option in Procedures that specifies to preload all the formats for the class variables.

The results will be as follows:
### Obs  trt     _TYPE_   _FREQ_   n
1  GrandTotal  1   20     20
2  MissingTreatment  1   0      0
3  Total1       1   14     14
4  Total2       1   14     14
5  Treatment A  1   6      6
6  Treatment B  1   8      8
7  Treatment C  1   6      6

**USING FORMAT TO STANDARDIZE CLINICAL DATA**

Often time in clinical trials, in order to summarize and report the data in the study, you need to standardize the original files like laboratory results.

Suppose you receive a Clinical Laboratory Data Standard file from a clinician as follows:

<table>
<thead>
<tr>
<th>LTYPE</th>
<th>LPARS</th>
<th>LPARL</th>
<th>ORES</th>
<th>NEWURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>URINE</td>
<td>UBLD</td>
<td>URINE BLOOD</td>
<td>2+</td>
<td>Positive/Abnormal</td>
</tr>
<tr>
<td>URINE</td>
<td>UBLD</td>
<td>URINE BLOOD</td>
<td>3+</td>
<td>Positive/Abnormal</td>
</tr>
<tr>
<td>URINE</td>
<td>UBLD</td>
<td>URINE BLOOD</td>
<td>1+</td>
<td>Negative/Normal</td>
</tr>
<tr>
<td>URINE</td>
<td>UPH</td>
<td>URINE PH</td>
<td>5~8</td>
<td>Negative/Normal</td>
</tr>
<tr>
<td>URINE</td>
<td>UPH</td>
<td>URINE PH</td>
<td>&gt;=8</td>
<td>Positive/Abnormal</td>
</tr>
<tr>
<td>URINE</td>
<td>EPIC</td>
<td>EPITHELIAL CELLS</td>
<td>2+</td>
<td>Negative/Normal</td>
</tr>
<tr>
<td>URINE</td>
<td>EPIC</td>
<td>EPITHELIAL CELLS</td>
<td>3+</td>
<td>Positive/Abnormal</td>
</tr>
<tr>
<td>URINE</td>
<td>EPIC</td>
<td>EPITHELIAL CELLS</td>
<td>1+</td>
<td>Negative/Normal</td>
</tr>
</tbody>
</table>

You can use the file above to create a SAS format to handle the standardization of laboratory data.

```sas
data work.labx;
  INFILE "C:\SAS\Data\Xls\labx.csv" MISSOVER DSD DLM=',' firstobs=2;
  length LTYPE LPARS LPARL ORES NRES $ 200;
  input LTYPE $ LPARS $ LPARL $ ORES $ NRES $;
run;
```

You can create a data set then convert the dataset into a format catalog by using Proc Format:

```sas
proc sql noprint;
  create table labfmt as
  select distinct trim(left(lpars))||\'_'||trim(left(ores)) as start,
    '\$labf' as fmtname,
    trim(left(nres)) as label
  from labx
  order by fmtname, start
  ;
quit;

data other;
```
FMTNAME='$labf';
START='Other';
LABEL='Unknown';
output;
run;
data fmtds;
set labfmt other;
run;
proc format cntlin=fmtds fmtlib;
run;

If you have the original subject laboratory data like this:

<table>
<thead>
<tr>
<th>Obs</th>
<th>USUBJID</th>
<th>VISIT</th>
<th>LTYPE</th>
<th>LPARS</th>
<th>LPARL</th>
<th>ORES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ABC11-USA-101-1001</td>
<td>1</td>
<td>URINE</td>
<td>EPIC</td>
<td>EPITHELIAL CELLS</td>
<td>2+</td>
</tr>
<tr>
<td>2</td>
<td>ABC11-USA-101-1001</td>
<td>2</td>
<td>URINE</td>
<td>EPIC</td>
<td>EPITHELIAL CELLS</td>
<td>1+</td>
</tr>
<tr>
<td>3</td>
<td>ABC11-USA-101-1001</td>
<td>1</td>
<td>URINE</td>
<td>UBLD</td>
<td>URINE BLOOD</td>
<td>1+</td>
</tr>
<tr>
<td>4</td>
<td>ABC11-USA-101-1001</td>
<td>2</td>
<td>URINE</td>
<td>UBLD</td>
<td>URINE BLOOD</td>
<td>2+</td>
</tr>
<tr>
<td>5</td>
<td>ABC11-USA-101-1001</td>
<td>1</td>
<td>URINE</td>
<td>UPH</td>
<td>URINE PH</td>
<td>5-8</td>
</tr>
<tr>
<td>6</td>
<td>ABC11-USA-101-1001</td>
<td>2</td>
<td>URINE</td>
<td>UPH</td>
<td>URINE PH</td>
<td>5-8</td>
</tr>
<tr>
<td>7</td>
<td>ABC11-USA-101-1002</td>
<td>1</td>
<td>URINE</td>
<td>EPIC</td>
<td>EPITHELIAL CELLS</td>
<td>3+</td>
</tr>
<tr>
<td>8</td>
<td>ABC11-USA-101-1002</td>
<td>2</td>
<td>URINE</td>
<td>EPIC</td>
<td>EPITHELIAL CELLS</td>
<td>1+</td>
</tr>
<tr>
<td>9</td>
<td>ABC11-USA-101-1002</td>
<td>1</td>
<td>URINE</td>
<td>UBLD</td>
<td>URINE BLOOD</td>
<td>1+</td>
</tr>
<tr>
<td>10</td>
<td>ABC11-USA-101-1002</td>
<td>2</td>
<td>URINE</td>
<td>UBLD</td>
<td>URINE BLOOD</td>
<td>3+</td>
</tr>
<tr>
<td>11</td>
<td>ABC11-USA-101-1002</td>
<td>1</td>
<td>URINE</td>
<td>UPH</td>
<td>URINE PH</td>
<td>5-8</td>
</tr>
<tr>
<td>12</td>
<td>ABC11-USA-101-1002</td>
<td>2</td>
<td>URINE</td>
<td>UPH</td>
<td>URINE PH</td>
<td>5-8</td>
</tr>
<tr>
<td>13</td>
<td>ABC11-USA-101-1003</td>
<td>1</td>
<td>URINE</td>
<td>EPIC</td>
<td>EPITHELIAL CELLS</td>
<td>2+</td>
</tr>
<tr>
<td>14</td>
<td>ABC11-USA-101-1003</td>
<td>2</td>
<td>URINE</td>
<td>EPIC</td>
<td>EPITHELIAL CELLS</td>
<td>4+</td>
</tr>
<tr>
<td>15</td>
<td>ABC11-USA-101-1003</td>
<td>1</td>
<td>URINE</td>
<td>UBLD</td>
<td>URINE BLOOD</td>
<td>3+</td>
</tr>
<tr>
<td>16</td>
<td>ABC11-USA-101-1003</td>
<td>2</td>
<td>URINE</td>
<td>UBLD</td>
<td>URINE BLOOD</td>
<td>2+</td>
</tr>
<tr>
<td>17</td>
<td>ABC11-USA-101-1003</td>
<td>1</td>
<td>URINE</td>
<td>UPH</td>
<td>URINE PH</td>
<td>&gt;=8</td>
</tr>
<tr>
<td>18</td>
<td>ABC11-USA-101-1003</td>
<td>2</td>
<td>URINE</td>
<td>UPH</td>
<td>URINE PH</td>
<td>&gt;=8</td>
</tr>
</tbody>
</table>

In order to statistically summarize the data, you want to standardize all laboratory results with above standard. With SAS formats you have already generated, you can simply do just this:

data lab1;
set labsubj;
length NRES FNAM $200 ;
FNAM = trim(left(lpars)) || '_' || trim(left(ores));
NRES = put(fnam, $labf.);
run;

Then, you will get the standardized data that is statistical-summary-ready.
CONCLUSION
The above examples shed a bit light on the great expectations of Proc Format. The Proc Format has a lot of uses in statistical summary, data process, output manipulation, and etc. The format techniques are simple but very useful if you can skillfully master the Proc Format and know your data characteristics.

ACKNOWLEDGMENTS
SAS is a Registered Trademark of the SAS Institute, Inc. of Cary, North Carolina.

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

- Author Name: John He, Ram Valluru
- Company: Octagon Research Solution
- Address: 585 Swedesford Ave, Wayne
- Work Phone: 610-535-6500
- Email: rvalluru@octagonresearch.com

*****************************************************************************