Paper BB-06 - 2015

Color, Rank, Count, Name; Controlling it all in PROC REPORT

Arthur L. Carpenter
California Occidental Consultants, Anchorage, AK

ABSTRACT

Managing and coordinating various aspects of a report can be challenging. This is especially true when the structure and composition of the report is data driven. For complex reports the inclusion of attributes such as color, labeling, and the ordering of items complicates the coding process. Fortunately we have some powerful reporting tools in SAS® that allow the process to be automated to a great extent.

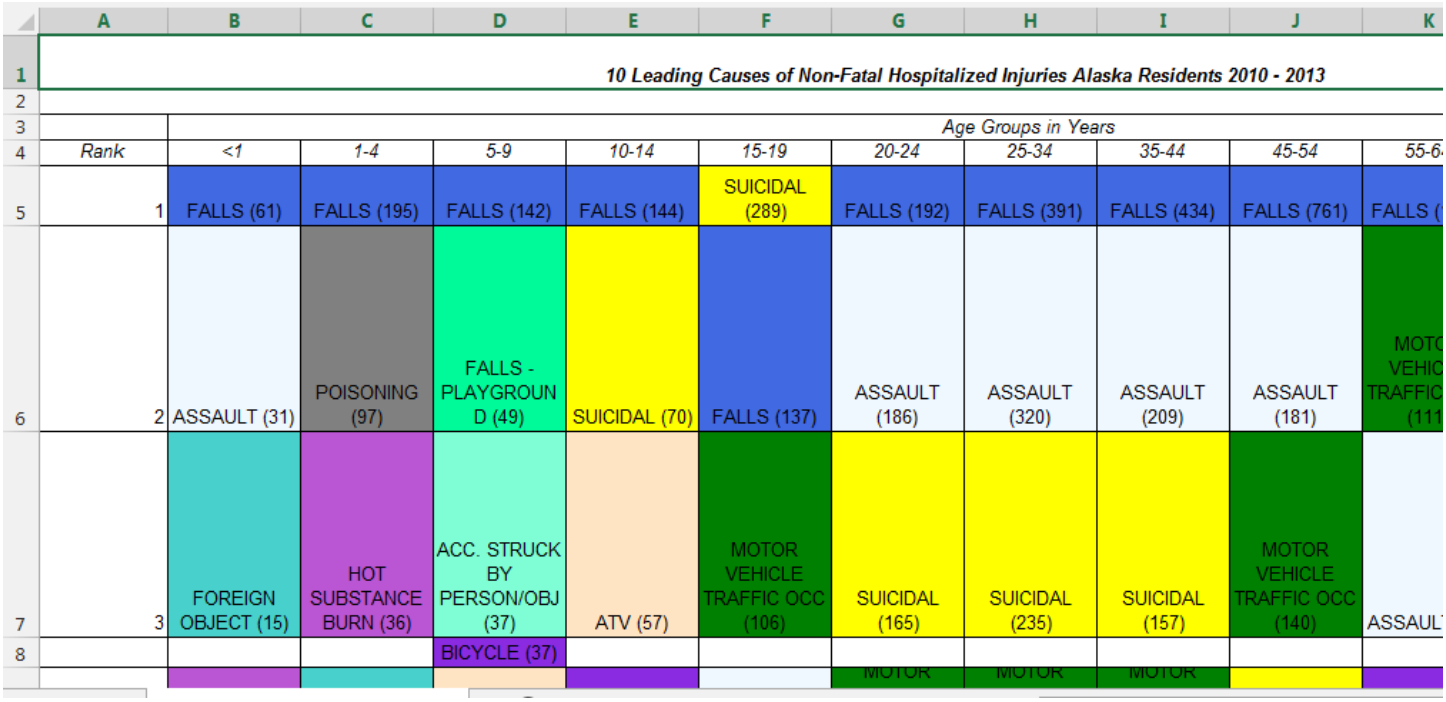
In the example presented in this paper we are tasked with generating an EXCEL® spreadsheet that ranks types of injuries within age groups. A given injury type is to receive a constant color regardless of its rank and the labeling is to include not only the injury label, but the actual count as well. Of course the user needs to be able to control such things as the age groups, color selection and order, and number of desired ranks.

KEYWORDS

PROC REPORT, PROC FORMAT, color selection, PROC RANK, user defined formats, traffic lighting

INTRODUCTION

Although the report that is to be generated initially appears to be fairly straightforward, there are a number of nuances that make the actual production of the report more interesting. For the purposes of this paper the final EXCEL spreadsheet is generated using PROC REPORT. Although the labels and color choices are those of the user of the SAS programs, the form of the report itself is dictated by the final consumer.



10 Leading Causes of Non-Fatal Hospitalized Injuries Alaska Residents 2010 - 2013

Rank	Age Groups in Years									
	<1	1-4	5-9	10-14	15-19	20-24	25-34	35-44	45-54	55-6
1	FALLS (61)	FALLS (195)	FALLS (142)	FALLS (144)	SUICIDAL (289)	FALLS (192)	FALLS (391)	FALLS (434)	FALLS (761)	FALLS (
2	ASSAULT (31)	POISONING (97)	FALLS - PLAYGROUND (49)	SUICIDAL (70)	FALLS (137)	ASSAULT (186)	ASSAULT (320)	ASSAULT (209)	ASSAULT (181)	MOTO VEHIC TRAFFIC (111
3	FOREIGN OBJECT (15)	HOT SUBSTANCE BURN (36)	ACC. STRUCK BY PERSON/OBJ (37)	ATV (57)	MOTOR VEHICLE TRAFFIC OCC (106)	SUICIDAL (165)	SUICIDAL (235)	SUICIDAL (157)	MOTOR VEHICLE TRAFFIC OCC (140)	ASSAUL
			BICYCLE (37)							

There are a number of challenges that have to be overcome to generate this report, while making it flexible enough for the user.

- The data can be grouped either within or across years and/or subsetted for specific sub-populations (such as rural trauma or by ethnicity).
- The final consumer of the report can select how many ranks to include ("report the top 10 trauma types across all age groups").
- Each cell must reflect a trauma description (which naturally is not a data value, but is a classification variable).
- The cell must also contain the number of traumas for that trauma type (counted by trauma type within age group).
- The color combinations must be tied to the trauma type so that ASSAULT, for instance, will always receive the same color regardless of what trauma types, data subsets, or number of ranks are selected.
- While the age groups are fairly fixed, they have unequal width and are not always present for all data subsets, although all age groups must be present for all reports.
- The user generating the report must be able to select and order the color choices and create an association between color and trauma type.
- **Each cell has a unique combination of position (age X rank), color, and text (trauma type and count)**

Clearly macro processing will be involved to generalize the subsetting and reporting process, however other aspects of the program are even more interesting. User defined formats and the RANK and REPORT procedures are used as well.

BRUTE FORCE APPROACH

In the original incarnation of the process that created the table, the user was generating the counts for the individual cells (trauma by age groups) using PROC SUMMARY and then ranking and transferring the information to Excel® by hand. This process was far too time consuming and error prone to be practical, especially when multiple data sets and various combinations of data subsets were requested.

Color selection was made by hand using the Excel color pallet, which provided flexibility but not consistency across reporting years. Also because the color selection was not by name or code (color selection was through the use of a color wheel), the colors were not always exactly the same from table to table.

Clearly we need to eliminate as much of the manual process as possible. What is needed is an automated way to build the report and write it directly into Excel.

ELEMENTS OF THE AUTOMATED APPROACH

The report itself is to be generated using PROC REPORT, and the output will be directed to Excel using the EXCELXP tagset. However because of the number of things that change at the cell level (background color, count, and trauma type), a less traditional approach was taken for the REPORT step.

Specifying User Defined Formats

Most of the control of the display attributes was specified through the use of user defined formats. The age groups were constant across reports and years, so the format for determining the age groupings was specified through the use of a VALUE statement, such as the one shown to the right. Here the various ranges of AGE values (in years) are mapped to corresponding labels.

```
proc format;
   value age_group
      .         = 'Unknown'
      low - 0   = 'Unknown'
      0< - <1   = '  <1'
      1  - <5   = ' 1-4'
      5  - <10  = ' 5-9'
. . . . some groupings not shown . . . .
      75 - <85  = '75-84'
      85 - high= '85+';
   run;
```

Most of the other user defined formats used to generate the report were generated during the execution of the program. These formats were built using control data sets which were based on the data, and were built during the report generation process.

If we were to build a format to map a code to a color using a VALUE statement, the code might look something like the PROC FORMAT step shown to the right. In this VALUE statement the numeric format BCOLOR is created with a series of numeric codes being mapped to specific colors. Any codes not specified in the list of codes falls into the OTHER category and is mapped to WHITE. If there are hundreds of codes or if the codes and colors were to depend on the data it would be impractical to physically code the VALUE statement. Fortunately if the CODE to COLOR pairings exist in a data set we can have the format generated for us – without typing the VALUE statement.

```
proc format;
   value bcolor
      11 = 'black'
      22 = 'red'
      33 = 'black'
      44 = 'red'
      other='white';
   run;
```

The data sets that are used to build user defined formats must have specific variables (Carpenter 2003 and 2015). The minimum three variables are FMTNAME, START, and LABEL. Often these types of formats will also utilize the HLO variable as well to control for special cases. These data sets will generally have a pairing of values that map a data value (START) with a text value (LABEL).

```
data codes;
   input code color $;
   datalines;
   11 black
   22 red
   33 black
   44 red
   run;
data control(keep=fmtname start label hlo);
   set codes(rename=(code=start color=label))
            end = eof;
   retain fmtname 'bcolor';
   output control;
   if eof then do;
      hlo='o';
      label='white';
      output control;
   end;
   run;
proc format cntlin=control;
   run;
```

In the example to the left the WORK.CODES data set simulates the CODE to COLOR pairings. This data set is used to create a new data set (WORK.CONTROL) with specific variables.

FMTNAME contains the name of the format for each observation.
START is the incoming data value that is to be mapped.
LABEL contains the color that is to be mapped to.
HLO is used to dictate which color will be assigned to all other values of CODE.

The CNTLIN= option on the PROC FORMAT statement is used to indicate that the data set has the necessary variables to build a format. The data set does not need to be named CONTROL and can contain multiple format definitions.

A number of data driven formats will be used to build the report. This is key because if some aspect of the report structure is changed by the user at execution time, the formats will self-adjust.

Report Layout

Because each cell of the report has a unique combination of display attributes, it was not possible to control the formatting of the individual cells directly. Instead the data were summarized outside of PROC REPORT. This allows us to build cell specific formats. The cells themselves were numbered consecutively. This allowed us to attach the attribute formatted values to the specific cells. Here cell #20 is tied with cell #19 (notice RANK is missing).

	Age Groups in Years									
Rank	<1	1-4	5-9	10-14	15-19	20-24	25-34	35-44	45-54	55-64
1	1	7	17	29	42	53	63	73	84	94
2	2	8	18	30	43	54	64	74	85	95
3	3	9	19	31	44	55	65	75	86	96
	.	.	20	.	.	.	.	.	.	.
4	4	10	21	32	45	56	66	76	87	97

Data Preparation Flow

In order to generate the unique cell numbers the data were first summarized, ranked, and then assigned cell numbers. Formats were then created based on the values of the individual cells.

1) Count the incidents within each age and trauma group using PROC FREQ. The data set (AGEFREQ) now has one observation for each injury type and age group. Each row will eventually become a cell in the report. The data set is sorted by age and trauma type, but we need them to be ranked by COUNT.

VIEWTABLE: Work.Agefreq (Frequency Counts and Percentages)

	age_years	et_ak1_as_text	COUNT
1	<1	ACC. STRUCK BY PERSON/OBJ	4
2	<1	AIRPLANE	2
3	<1	ASSAULT	31
4	<1	CAUGHT BETWEEN OBJECTS	1
5	<1	FALLS	61

2) PROC RANK is used to rank the trauma types by the number of instances. Only the top &NUMRANK ranks are returned by PROC RANK and these values are written to the data set AGERANKS. The variable CNTRANKS will be used to order the rows in the table.

```
proc rank data=agefreq
          out=ageranks(where=(cntrank le &numrank))
          descending ties=dense;
   by age_years;
   var count;
   ranks cntrank;
   run;
```

VIEWTABLE: Work.Ageranks

	age_years	et_ak1_as_text	COUNT	cntrank
1	<1	FALLS	61	1
2	<1	ASSAULT	31	2
3	<1	FOREIGN OBJECT	15	3
4	<1	HOT SUBSTANCE BURN	14	4

3) Each row is a cell and each cell needs a unique identifier. An easy way to do this is to assign a numeric counter (START). This can be done easily in a DATA step. We will also need to have a variable to identify the row within each rank in the table (ROW). This is necessary because there may be ties in the number of traumas within an age group. In this same DATA step we can start to build the formats used to assign the attributes to the individual cells. START is used as the name of the cell identifier, because PROC FORMAT requires the use of a variable with that name. In this same DATA step we can build the control data sets that will be used to create the user defined formats that control the cell attributes.

```
if first.cntrank then row=0;
* Row within rank;
row+1;
* Create a unique cell id;
start+1;
cstart=left(put(start,5.));
output freq;
```

VIEWTABLE: Work.Freq

	age_years	et_ak1_as_text	cntrank	row	start
1	<1	FALLS	1	1	1
2	<1	ASSAULT	2	1	2
3	<1	FOREIGN OBJECT	3	1	3
4	<1	HOT SUBSTANCE BURN	4	1	4
5	<1	SUFFOCATION	5	1	5
6	<1	POISONING	6	1	6
7	1-4	FALLS	1	1	7
8	1-4	POISONING	2	1	8
9	1-4	HOT SUBSTANCE BURN	3	1	9
10	1-4	FOREIGN OBJECT	4	1	10

4) The control files used to create the attribute formats will have the variables: FMTNAME, START, LABEL, HLO, and MAX. At this point two formats are to be created. One to control the background color (FMTNAME='$tablecolor') and the other the text label (FMTNAME='tablelbl') to be displayed within each cell.

```
data freq(keep=cntrank row age_years start)
     controlclr(keep=fmtname cstart label hlo max
                rename=(cstart=start))
     controllbl(keep=fmtname start label hlo max);
   set ageranks end=eof;
```

```
label =  catt(&freqvar,' (',left(put(count,8.)),')');
```

The text used in the cell includes the trauma type, as well as the count (COUNT), consequently the value of two variables are combined in the display text (LABEL) for the TABLELBL. format. The unique cell identifier is stored in START. Here the last few lines of the CONTROLLBL data set are shown. The HLO='o' allows unspecified values of START to be mapped to a blank label.

VIEWTABLE: Work.Controllbl

	label	max	start	fmtname	hlo
163	OTHER ROAD VEHICLE (1)	55	163	tablelbl	
164	POISONING (1)	55	164	tablelbl	
165	SNOW MACHINE (1)	55	165	tablelbl	
166	STRAIN (1)	55	166	tablelbl	
167	SUFFOCATION (1)	55	167	tablelbl	
168	WATER TRANSPORT W/O DROWN (1)	55	168	tablelbl	
169		55	168	tablelbl	o

Because of the requirement to allow the user to select color combinations as well as to assure that a given trauma type will always have the same color, regardless of whether or not that trauma type appears in a particular table, mapping the background color for the cell is more convoluted. First the $INJ_GRP_RNK. format is used to map the trauma type (&FREQVAR) to a code associated with a background color. This mapping is used to create the $TABLELBL. format, which in turn maps the cell number to the color code. The ability to build a list of colors and to use them by name is discussed in Carpenter (2015).

```
label =  put(&freqvar, $inj_grp_rnk.);
```

VIEWTABLE: Work.Controlcolor

	label	fmtname	start	hlo
1	RoyalBlue	$FVColor	1	
2	AliceBlue	$FVColor	2	
3	Yellow	$FVColor	3	
4	Green	$FVColor	4	
5	Aquamarine	$FVColor	5	
6	Bisque	$FVColor	6	
7	BlueViolet	$FVColor	7	

5) The TABLELBL. and $TABLECLR. formats are used in the PROC REPORT step that generates the final table.

```
proc report data=freq nowd  out=seerpt;
   column ('Rank' cntrank) row age_years,start dummy; ❶
   define cntrank / group ' ' ; ❷
   define row     / group noprint; ❸
   define age_years / across order=formatted 'Age Groups in Years'; ❹
   define start /  analysis ' ' sum center ❺
                   f=tablelbl55.
                   style(column) = {cellwidth=70pt cellheight=90pt};
   define dummy / computed noprint ; ❻
   compute dummy;
      length color $35;
      * Array index will be the column numbers;
      array strt (3:15) _c3_ _c4_ _c5_ _c6_ _c7_ _c8_ _c9_  ❼
                        _c10_ _c11_ _c12_ _c13_ _c14_ _c15_;
      do i = 3 to 15; ❽
         * format with a color;
         if strt{i}=. then color='White';
         else color= put(put(left(put(strt{i},5.)),$tablecolor5.),$fvcolor.); ❾
         if color=' ' then color='white';
         call define(i,'style','style={background='||color||'}'); ❿
      end;
   endcomp;
   run;
```

❶ Report items are defined. DUMMY only exists to support the compute block.

❷ CNTRANK is the rank number and is used to form rows in the table.

❸ ROW is the actual row number which is needed to allow ties (two rows for one value of CNTRANK.)

❹ Age in years has a define usage of ACROSS. Each age group will be used to form a column. These will be columns #3 through #15. Although not discussed in this paper, the program insures that all age groups will appear.

❺ START is the unique cell identifier. Since it is nested within an across variable, it will appear in each cell. PROC REPORT requires that this variable be numeric.

❻ DUMMY only exists so that we can have a compute block that will be executed after all the other columns. DUMMY does not appear in the report.

❼ When AGE_YEARS is transformed, it will form columns 3 through 15. An array is used to hold the report value (unique cell number) for each of these columns.

❽ A DO loop is used to step through the elements of the array.

❾ The cell number is translated into the color number using the $TABLECOLOR. format. The color number is in turn translated into a color name using the $FVCOLOR. format., and this color name is used to specify the background color.

❿ The CALL DEFINE is used to change the background color using the temporary variable COLOR for this specific cell.

6) The PROC REPORT step resides inside of an ODS sandwich. In this case ODS is used to build a excel workbook for delivery to the client.

```
ods tagsets.excelxp
      file="&path\reports\Agefreq_&acctype._&restype._&yrstrt._&yrstop..xls"
      style=journal
      options(embedded_titles='yes'
              embedded_footnotes='yes'
              Orientation='Landscape');

. . . . code not shown . . . .
proc report data=freq . . . .;
. . . . code not shown . . . .

ods tagsets.excelxp close;
```

i

ables used in the naming of the file are supplied by the user when the macro that drives the report process is called.

A FEW MACRO SPECIFICS

Adding flexibility to any program must necessarily add complexity as well. For users unfamiliar with complex SAS code, the macro language can be used to provide this flexibility to the user. In this particular application the user needed the flexibility to be able to select from a variety of input parameters.

```
%macro agefrequencymap(yrstrt=11, yrstop=12,FreqVar=inj_ecode01,
                       numrank=10, mincnt=5, other=no, unknown=no,
                       acctype=NONFATAL, restype=AK);
*********************************;
*
* The start and stop years together determine the incoming data set
* Parameter default     alternate values and description
* yrstrt    11          two digit start year
* yrstop    12          two digit stop year
* freqvar   inj_ecode01 analysis variable of interest
* numrank   10          Number of ranks to include in the table/graphic
* mincnt     5          Minimum number of frequency values to be included in the chart
* other     no          no       do not include the OTHER category in the chart
                        yes      OTHER can appear in the chart
* unknown   no          no       do not include the UNKNOWN category in the chart
                        yes      UNKNOWN can appear in the chart
* acctype   nonfatal    nonfatal exclude fatal injuries
*                       fatal    include only fatal injuries
*                       all      include all injuries (fatal or non-fatal)
* restype   AK          AK       Alaska residents only
*                       other    non-alaska residents only
*                       all      no residency exclusions
*************************************************************************;
```

These included the ability to summarize over multiple years, change the analysis variable of interest, select the size of the table by limiting the number of ranks (rows), and other selection criteria. Throughout all these combinations the tables will have a uniform look. A FALL will always have the same shade of blue whether it is ranked 1 or 25.

SUMMARY

PROC REPORT has the ability to format a complex array of report types. When combined with user defined formats, it is possible to gain and maintain extensive control over the report process and the appearance of the final report. User defined formats especially those generated on the fly based on the data being used in the analysis provides an extra degree of automation.

In the report generated in this paper each individual cell of the report was assigned a unique value. A series of formats were then generated that tied that unique value to corresponding attributes for that cell. The result total control of the report at the cell level.

ABOUT THE AUTHOR

Art Carpenter's publications list includes; five books, two chapters in *Reporting from the Field*, and numerous papers and posters presented at SAS Global Forum, SUGI, PharmaSUG, WUSS, and other regional conferences. Art has been using SAS since 1977 and has served in various leadership positions in local, regional, and national user groups.

Art is a SAS Certified Advanced Professional Programmer, and through California Occidental Consultants he teaches SAS courses and provides contract SAS programming support nationwide.

Recent publications are listed on my sasCommunity.org Presentation Index page. SAS programs associated with this paper can be found at:
http://www.sascommunity.org/wiki/Using_Arrays_to_Quickly_Perform_Fuzzy_Merge_Look-ups_Case_Studies_in_Efficiency

AUTHOR CONTACT

Arthur L. Carpenter
California Occidental Consultants
10606 Ketch Circle
Anchorage, AK 99515

(907) 865-9167
art@caloxy.com
www.caloxy.com

REFERENCES

Carpenter, Arthur L., 2003, "Building and Using User Defined Formats", Proceedings of the 11th Annual Western Users of SAS Software, Inc. Users Group Conference, Cary, NC: SAS Institute Inc. Also in the proceedings of the Twenty-ninth SAS User Group International Conference (SUGI), 2004, Cary, NC: SAS Institute Inc., also in the proceedings of the Mid West SAS User Group Conference (MWSUG), 2005, Cary, NC: SAS Institute Inc. Also presented at the 16th Annual Western Users of SAS Software, Inc. Users Group Conference (WUSS), Universal City, CA in 2008 and as a hands-on workshop at the 2015 WUSS conference in San Diego.

Carpenter, Arthur L., 2015, "Controlling Colors by Name; Selecting, Ordering, and Using Colors for Your Viewing Pleasure". Presented at the 2015 Midwest SAS Users Conference.

TRADEMARK INFORMATION