Tricks to Quickly Analyze and Graphically Represent Numeric Data
Anastasiya Osborne, Farm Service Agency (USDA), Washington, DC

ABSTRACT
Have you ever been asked to produce a high quality, management-friendly report in record time? Have you ever spent time typing ranges for PROC FORMAT to apply in tables or maps? During Congressional hearings, U.S. Department of Agriculture (USDA) often gets urgent requests to graphically represent politically-sensitive data. This paper presents a SAS® macro that was developed to allow flexibility in choosing a dataset, a variable in question, and a number of groups for statistical analysis. The macro then produces the results in an Excel spreadsheet, and an ODS output. It also automatically creates a format for the variable that can be used in PROC GMAP to produce an impressive map. The macro reduces programming time by eliminating time-consuming tasks to analyze the variable and manually type ranges for PROC FORMAT.

INTRODUCTION
Being a member of the Economic and Policy Analysis Staff at the Farm Service Agency (FSA), USDA requires stamina and creativity. A stream of urgent requests to produce ad-hoc reports with statistical analysis of data can come at any time. The effort in creating these reports can be time-consuming and inefficient, especially when analysis of unfamiliar data is needed within a short period of time, as, for example, during Congressional deliberations. This is when SAS MACRO facilities can be handy. MACRO saves time and automates a tedious mistake-prone process of typing format ranges, so that the mind of the analyst is freed to tackle more complicated issues. This automated approach to analyze the variable, create a user-defined format, and map data drastically reduces staff time to produce a report.

WHY AUTOMATE THE MAP-MAKING PROCESS?
Our office prepares analytical reports for FSA and USDA senior management, and sometimes for Congressional staff. Automating the report writing process as much as possible helps to provide consistent data analysis, include frequently requested statistics, present data in a standard format, and reduce turn-around time of multiple and often similar requests. One conventional statistical technique for creating useful maps is to divide the range of the variable into several groups - for example, 5, 7, 10, and so on.

The analyst would decide on the number of groups, and SAS would create those groups with similar frequencies. If the range is too large, as in the following example with U.S. corn production by county, it is hard to divide the range (e.g. from 500 bushels to 58,775,267 bushels) into equal parts and create a meaningful map. However, mapping frequencies, rather than evenly distributed ranges, creates a much more user-friendly product.

The macro that generates these reports uses three distinct SAS programming techniques: 1) generate ranges for each variable based on a user-requested number of frequency groups; 2) assign ranges to macro variables for use in PROC FORMAT 3) use the new format in SAS/GRAPH procedures (e.g. PROC GMAP). The resulting code will automatically generate a colorful standard map showing a user-requested number of groups, for any variable. The code also generates an Excel file for documentation purposes.

GENERATING RANGES
First, we need to look at the data. To create “best” ranges for use in PROC FORMAT, first use PROC FREQ to examine the data we intend to map.

```
PROC FREQ data = &DS noprint;
   tables &VAR/missing norow nocol nopercent
   out = freq_&VAR
   ;
RUN ;
```
If we have many variables (possibly in multiple datasets), we can use a macro to look at them.

```sas
%M macro MYFREQ ( DS = , VAR = ) ;
   PROC FREQ data = &DS noprint;
      tables &VAR/missing norow nocol nopercent ;
      out  = freq_&VAR
   ;
   RUN ;
%M mend MYFREQ ;
%M MYFREQ (DS = DS, VAR = VAR1 ) ;
%M MYFREQ (DS = DS, VAR = VAR2 ) ;
...  
%M MYFREQ (DS = DS, VAR = VARn ) ;
```

Let's say we have one variable. A careful look at the frequency dataset will guide us in choosing the number of groups we will use.

**USING PROC RANK/SORT/SUMMARY/EXPORT TO GROUP AND DOCUMENT THE DATA**

Then we assign a macro parameter – how many groups are needed. The usual convention is to group data into 100 - for percentiles, 10 - for deciles, and 4 - for quartiles. For my work, we usually find that 5 to 10 groups are about right for a readable map. If the range is very large (see the following map for U.S. corn production) and “lumpy” (i.e. there are large gaps in the data or clusters around certain values), we may need more groups to illustrate the data properly.

For example, **deciles** would be assigned using the following code:

```sas
PROC RANK groups=10 data=s_&DS
   out= rank Prepare ties=high ;
   var &VAR;
      ranks rank_&VAR ;
RUN ;
```

The rank-prepare dataset needs to be sorted.

```sas
PROC SORT data = rank_prepare ;
   by rank_&VAR ;
RUN ;
```

Then we use PROC SUMMARY to find the minimum and maximum of each group:

```sas
PROC SUMMARY data=rank_prepare n max ;
   var &VAR;
   by Rank_&VAR ;
   output out= minmax&VAR
      min =min&VAR._byrank
      max =max&VAR._byrank
   ;
RUN ;
```
We export results to a user-friendly Excel file for documentation:

```sas
/* NOTE - use your preferred location */
PROC EXPORT data = group&VAR
   outfile = "H:\SAS programs for 2008\&dataset &lryear groups for &VAR..xls"
   dbms = excel2000
   replace ;
RUN ;
```

Historically, the next step here has been to use a CALL SYMPUT to create the format. However, later on another way was developed to accomplish the same goal. The following section describes the historic way.

**USING CALL SYMPUT TO ASSIGN RANGE VALUES TO MACRO VARIABLES**

With CALL SYMPUT, we create a macro for the min and max range of each group. The macro variables will have the same name as the dataset variables, and have an additional suffix for the rank number.

```sas
DATA doit ;
set group&VAR ;
suffix = normal_rank&VAR ; /* We need a suffix to create macro variables out of each min and max value for each rank */
array xxx(*) _numeric_ ; /* "*" is needed to count the number of variables */
do i = 1 to dim(xxx);
call symput(cats(VNAME(xxx[i]), suffix), xxx[i]) ; /* CALL SYMPUT is a DATA step subroutine the assigns a value to the macro variable. VNAME assigns the name of the variable A as the value of the variable B. CATS is a concatenation function to add a suffix to the name supplied by VNAME function. */
end ;
RUN ;
```

This is how the log looks.

```markdown
MPRINT(MKFMT):   VALUE CORpro2008f 500 - 5,087 = "500 to 5,087" 5,133 - 12,398 = "5,133 to 12,398" 12,657 - 38,725 = "12,657 to 38,725" 39,381 - 125,000 = "39,381 to 125,000" 125,333 - 327,012 = "125,333 to 327,012" 327,667 - 801,000 = "327,667 to 801,000" 803,933 - 2,033,333 = "803,933 to 2,033,333" 2,048,000 - 5,015,667 = "2,048,000 to 5,015,667" 5,036,667 - 12,071,267 = "5,036,667 to 12,071,267" 12,160,000 - 58,775,267 = "12,160,000 to 58,775,267" ;
NOTE: The previous statement has been deleted.
MPRINT(MKFMT):   RUN ;
```

You then have to copy this text and paste it into the body of the SAS program.

**USING CNTLIN OPTION TO CREATE A FORMAT**

This is a way to accomplish the same goal without having to copy and paste a part of the log manually into the SAS program, although in the previous method it is easier to spot problems with the underlying data (e.g., not enough data to make the desired number of groups).
Use the CNTLIN option in PROC FORMAT to create a format from a dataset automatically. We need a dataset with the variables START (values), LABEL (descriptions for those values), and FMTNAME (how we want to name this format). FMTNAME is a character variable and requires quotes.

```plaintext
DATA
fmtdata_&VAR ( keep = fmtname start end label)
group&VAR ( keep = normal_rank&VAR start end _freq_
rename = (start=min&VAR end=max&VAR) )
length normal_rank&VAR start end 8. fmtname $ 32
label $ 32;
retain fmtname "&VAR.f" ; /* This will be repeated in each row */
set minmax&VAR end = EOF;
by rank_&VAR;

normal_rank&VAR = rank_&VAR + 1; /* We see that the ranks start at zero. To create a “normal rank”, I add 1. */
if normal_rank&VAR = . then delete;
start = round(min&VAR.by_rank, 0.01);
end = round(max&VAR.by_rank, 0.01);
label = catx( " " , put(start, comma14.0 ) , "to" , put(end, comma14.0) ) ;
if eof then
call symputx ( “last_rank’ , normal_rank&var ) ;
RUN ;

To use the CNTLIN option in PROC FORMAT, comment out the previous DATA doit step and run the following lines calling PROC FORMAT with the CNTLIN option.

PROC FORMAT cntlin = fmtdata_&VAR ;
RUN ;
PROC FORMAT fmtlib ;
RUN ; /* For printing the contents of a format catalog */

PROC FORMAT USING THE MACRO
Now lower and upper limits of each group become numbered macro variables. The number of ranks is the value for the macro variable &last_rank.

%macro MKFMT ( ) ;
%local i ;
PROC FORMAT;
VALUE &VAR.f
%do i = 1 %to &last_rank ;
&&min&VAR&i - &&max&VAR&i = "&&min&VAR&i to &&max&VAR&i"
%end ;
RUN ;
%mend MKFMT ;

options mprint ;
%MKFMT()
This is what we get looking at the fmtdata_&var dataset.

<table>
<thead>
<tr>
<th>Obs</th>
<th>start</th>
<th>end</th>
<th>fmtname</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500</td>
<td>5087</td>
<td>CORpro2008f</td>
<td>500 to 5,087</td>
</tr>
<tr>
<td>2</td>
<td>5133</td>
<td>12398</td>
<td>CORpro2008f</td>
<td>5,133 to 12,398</td>
</tr>
<tr>
<td>3</td>
<td>12657</td>
<td>38725</td>
<td>CORpro2008f</td>
<td>12,657 to 38,725</td>
</tr>
<tr>
<td>4</td>
<td>39381</td>
<td>125000</td>
<td>CORpro2008f</td>
<td>39,381 to 125,000</td>
</tr>
<tr>
<td>5</td>
<td>12533</td>
<td>327012</td>
<td>CORpro2008f</td>
<td>125,333 to 327,012</td>
</tr>
<tr>
<td>6</td>
<td>327667</td>
<td>801000</td>
<td>CORpro2008f</td>
<td>327,667 to 801,000</td>
</tr>
<tr>
<td>7</td>
<td>803933</td>
<td>2033333</td>
<td>CORpro2008f</td>
<td>803,933 to 2,033,333</td>
</tr>
<tr>
<td>8</td>
<td>2048000</td>
<td>5015667</td>
<td>CORpro2008f</td>
<td>2,048,000 to 5,015,667</td>
</tr>
<tr>
<td>9</td>
<td>5036667</td>
<td>12071267</td>
<td>CORpro2008f</td>
<td>5,036,667 to 12,071,267</td>
</tr>
<tr>
<td>10</td>
<td>12160000</td>
<td>58775267</td>
<td>CORpro2008f</td>
<td>12,160,000 to 58,775,267</td>
</tr>
</tbody>
</table>

One way or the other, we are now ready to map the data.

**USING DEFINED FORMAT IN SAS/GRAPH**

Before using the defined format in PROC GMAP, it is good practice to set GOPTIONS for an appealing look of the map. Here is the recommended page setup, including legend options:

```
GOPTIONS RESET = GOPTIONS /*global*/

GUNIT = pct /* specifies the unit as percent, unless explicitly specified in another SAS statement. */

border /* Printed map is framed */
cback = white
ROTATE = landscape /* The way it's printed on a page */
GSFMODE = append /* This is to control whether maps replace one another, or append to the bottom of the specified file */
Ctext = black
Ftext = swiss
Htext = 9 PT ;
TITLE2 f = swissb ; /* 'Times-Bold'; */

legend1 mode = RESERVE /*Better than SHARE – no map crowding occurs when the legend is partly covered by the map.*/
shape = bar(1.4, 1.6) /* Shape of the shaded bars. Units specified in GUNIT option. */
position = (top) /*(bottom)*/ /* POSITION moves the legend above the map. Defaults: outside, center. */
value = (font = swiss)
label = ('') /* LABEL used to suppress the legend label. Default: The name of the response variable.*/
frame /* Box drawn around the legend */
cshadow = gray9a /* This adds a drop shadow to the box around the legend*/
cborder = gray
cframe = white;
```
I also developed a color scheme. There is SAS documentation about the colors, but it takes some time to develop a scheme, as the colors look different on different printers.

There can be different schemes depending on what is being depicted. For example, maps showing changes in payments range in color from light yellow to dark green. A scheme for changes in loan rates comparative to the last year takes gray as the "No change" color, green-blue-black for negative changes, and yellow-pink-red for positive changes. Below is a color scheme for crop production:

/* Color patterns for crop production */
pattern1 v = msolid c = LemonChiffon;
pattern2 v = msolid c = cxffcc00; /* gold */
pattern3 v = msolid c = orange;
pattern4 v = msolid c = cxd9892b; /*brilliant orange */
pattern5 v = msolid c = brown;
pattern6 v = msolid c = cx6ea688; /* light green */
pattern7 v = msolid c = green; /* green */
pattern8 v = msolid c = h03ccccf; /* pinkish violet, beautiful color */
pattern9 v = msolid c = cx9b58a6; /* light purple */
pattern10 v = msolid c = cx991f85; /*deep purplish pink */

MAP MAKING

Now we use the created format to generate a map of the variable. I have a master mapping program where I use %INCLUDE and call the macro %ASSIGN to create formats. There are better ways to do this, such as using the AUTOCALL macro facility, which permits faster updates and better SAS program consistency (see Jensen). %INCLUDE is what I am using here.

This macro %ASSIGN is run in another macro, %GET_FORMATS, if we need to map many crops at a time.

%macro GET_FORMATS (DS = , VAR = , GROUP = , CR = , CROP = , unit = ) ;
.../*create a subset if the master database, sort it by the variable */
%INCLUDE "H:\SAS papers - mine\assign groups June 2 2008 new.sas";
%ASSIGN (DS = &CR._formap, VAR = &CR.pro&lryear, GROUP = &GROUP, 
CR = &CR, CROP = C&CROP, unit = &UNIT ) ;
%mend GET_FORMATS ;

%GET_FORMATS (DS = COR_formap, VAR = CORpro&lryear, GROUP = 10,
CR = COR, CROP = Corn, unit = Production (bushels));

In the "Using Proc Rank/Sort/Summary/Export to Group the Data" section, I explained what is going on behind the scenes when %GET_FORMATS is run.

Now we are all set up for mapping. My group prefers to have a PDF file as the output on a shared drive where all maps for a given project are located:

/* OUTPUT */
filename OUT "H:\31_Maps\Output of maps lr2008\&VAR 2002-2006 OAP.pdf";

ODS listing close;
ODS PDF file= OUT ;
title1 h= 15 PT f = swissb "&VAR: 2002-2006 Olympic Average Production &unit";
PROC GMAP data = &VAR._formap map = maps.uscounty;

id state county; /* These are the variables common to the "maps" dataset and the current dataset we are trying to map. ID statement declares the variables that match response data set observations to map dataset areas. Second ID means that the map will display the boundaries of all state map areas, but only those county map areas that are requested. */

choro &VAR /*This is the requested variable we have been trying to map all this time. CHORO statement instructs PROC GMAP to create a choropleth map displaying values of the variable */
annotate = HOME.slines /* This way we don't see the separations between counties - we see a group of counties with the same color as one block */
cempty = white /* Specifies that empty areas are to be outlined in white color */
coutline = same
discrete
legend = legend1;

note height = 10 pt f = swissb move = (3,5) color = black 'Economic and Policy Analysis Staff, FSA/USDA';
note height = 10 pt f = swissb move = (3,3) color = black 'Data comes from NASS/USDA';

format &VAR &VARF.; /* Here is our format. */

RUN;
QUIT;

title; /* TITLE and FOOTNOTE statements are global and can appear anywhere within SAS code. They are placed in their own areas within the graphics output. They decrease the area map. No overwriting of the map occurs.*/

GOPTIONS RESET = pattern; /* To cancel all current PATTERN statements, use the RESET = option in GOOPTIONS statement */

ODS PDF close;
HOW IT LOOKS
You can see the deep-purple "Corn Belt" – the area in the Midwest of the United States where corn is the predominant crop. The data is publicly available and comes from the National Agricultural Statistics Service (NASS). Iowa, Illinois, Indiana, and Ohio grow about 50% of U.S. corn.

CONCLUSION
As you can see, these tricks help you automate a format creation and a map-making process. Automation of a report writing helps reduce errors and speeds up the report turn-around time. Hopefully you will find these tricks helpful.
REFERENCES


ACKNOWLEDGMENTS
Special thanks to Ian Whitlock for teaching me SAS in 2000 and up to this day, and to Dalia Kahane for moral support in writing this paper.

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

Anastasiya Osborne
Farm Service Agency, USDA
1400 Independence Ave, S.W.
Washington, DC 20250-0506
Work Phone: 202-690-0446
E-mail: stefan.osborne@comcast.net

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 trademarks of their respective companies.