

Paper RIV74

How to Make a Stunning State Map Using SAS/Graph® for Beginners

Sharon Avrunin-Becker, Westat, Rockville, MD

ABSTRACT

Making a map for the first time can be an overwhelming task if you are just beginning to learn how to navigate your way through SAS/Graph. It can be especially paralyzing when you are trying to narrow your map to a smaller scale by identifying counties in a state. This paper will walk you through the steps to getting started with your map and how to add ranges of colors and annotations. It will also point out a few traps to avoid as you are designing your programs and maps.

INTRODUCTION

In great anticipation of the 7th Star Wars movie release, a fictional researcher has decided to continue tracking the characteristics of devout Star Wars fans. He has developed a survey and has posted it on various fan based sites. He wants to track how many fans have completed the survey in the state of Maryland based on gender.

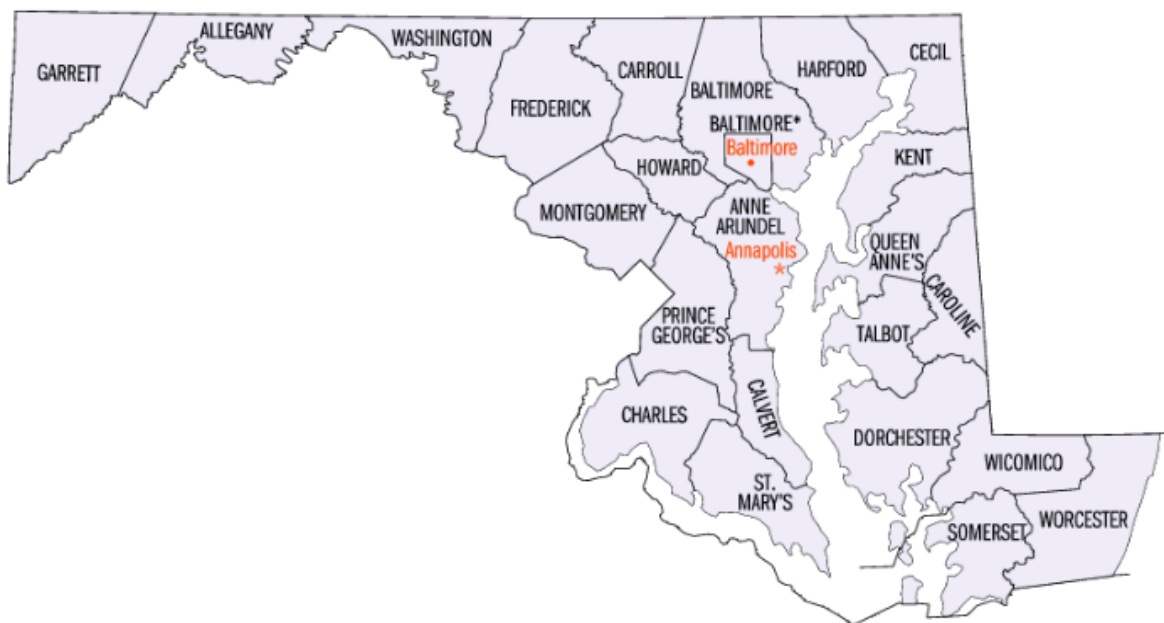


Figure 1. Example of a map of Maryland not done with SAS

HOW TO GET STARTED

SAS provides datasets that contain the information for SAS to know where to draw the boundary lines for counties, states, and countries. These datasets can be viewed by clicking in the Explorer window under Libraries and then clicking on the Maps library. By referencing these datasets you have the basics to begin to tell SAS what kind of map you are trying to draw.

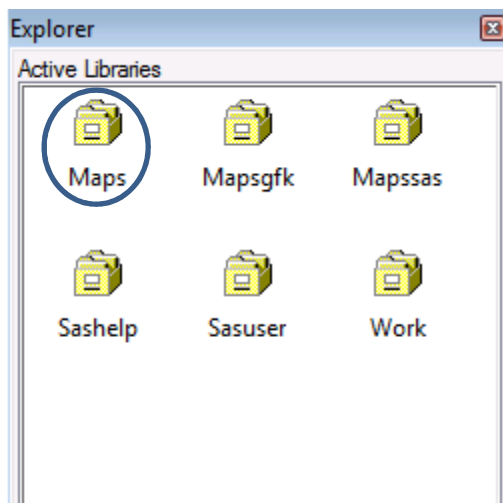


Figure 2. Location of Maps directory

GET COUNTS BY COUNTY

Now that we know where to find the information for the maps, let's get the data ready to be used. First, we are going to get counts by the counties and genders. I have broken this into two separate SQL steps so the beginner programmer can see it broken down

```
proc sql;
  create table femalecounty as
  select county , count(*) as cnt
  from x
  where gender='F'
  group by county ;

  create table malecounty as
  select county , count(*) as cnt
  from x
  where gender='M'
  group by county ;
quit;
```

FIRST ATTEMPT TO CREATE A MARYLAND MAP

The type of maps we are creating are called choropleth maps. According to Wikipedia, choropleth maps are based on statistical data aggregated over previously defined regions (e.g., counties). We produce the first drawing of our map by using PROC GMAP and pulling from the counties dataset from the Maps library which contains the X and Y coordinates for the counties in Maryland. After looking up State FIPS codes, we know to limit the state to FIPS code 24 to get Maryland counties.

```
proc gmap
data=maps.counties
map=maps.counties;
id county;
choro county;
where state eq 24;
run;
quit;
```

HERE IS THE OUTPUT...

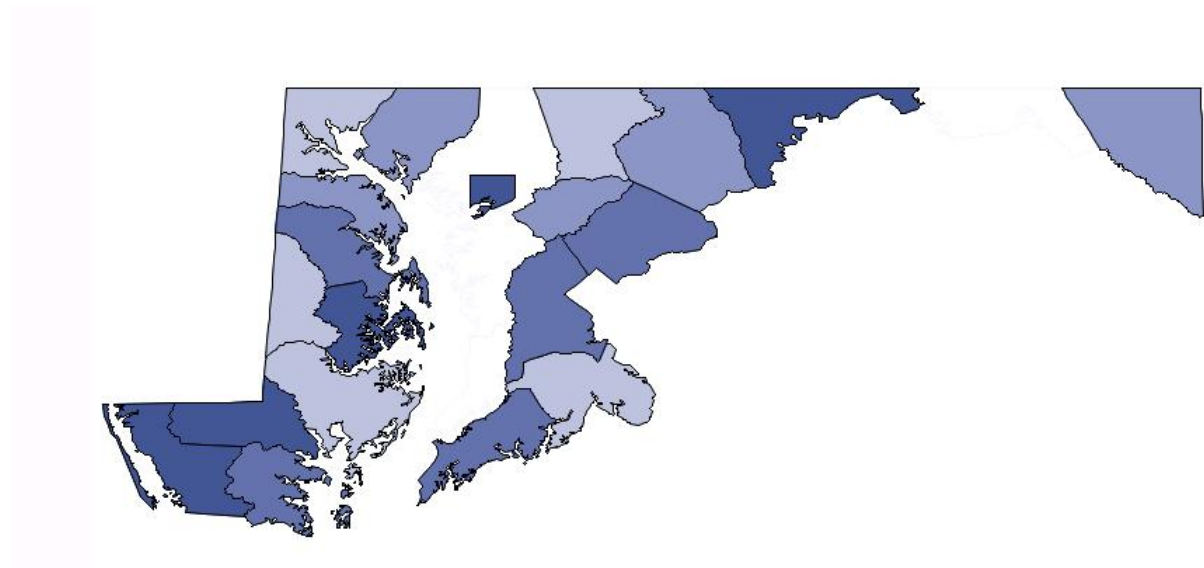


Figure 3. First attempt at our map

SECOND ATTEMPT TO CREATE A MARYLAND MAP – PROC GPROJECT

Oh Bantha Breathe!! Our map is backwards! What happened? What happened is that the longitudinal and latitudinal coordinates are in a spherical format and we need them to be in a Cartesian format. SAS has a procedure called PROC GPROJECT that converts the spherical coordinates to a flat Cartesian format. PROC GPROJECT does not produce the output map, but instead becomes the input data set that PROC GMAP uses to produce a correctly plotted map.

```
proc gproject
data=maps.counties
out=MDCounties;
where state eq 24;
id state county;
run;
```

```
proc gmap
data=MDCounties
map=MDCounties
all;
id county;
choro county;
run;
quit;
```

HERE IS THE OUTPUT...

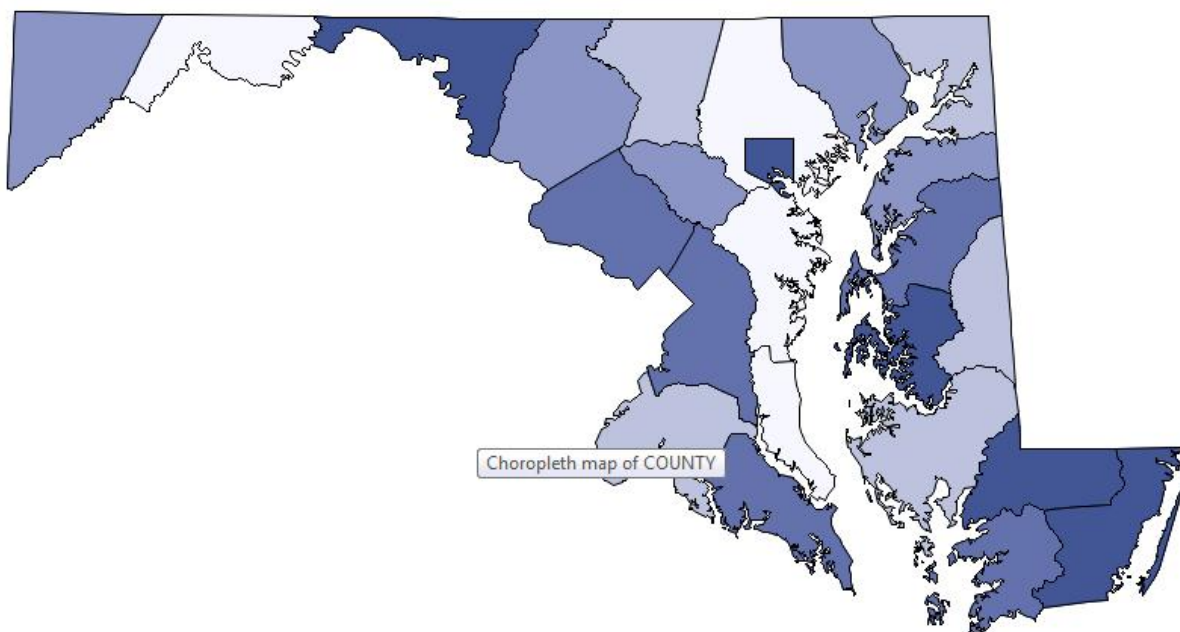


Figure 4. Map using PROC GPROJECT

ADDING IN OUR DATA

Now we are ready to add in our data. First we will merge in a fake data set that contains all the counties so if a county is missing we can set it to zero and we will add another variable that will identify some data ranges to be mapped. Formatting alone could also be used to set the ranges, but in this case we went with adding a variable called CNTRNG to help the new recruits out in following what is being demonstrated. A variable called XTEXT has also been created which will later be used for the annotations.

```
proc format ;
value genrng;
1 = '<5'
2 = '5 - 10'
3 = '11 - 20'
4 = '21 - 30'
5 = '>30'
;
quit;

data male;
length xtext $30.;
merge malecounty
      fakeit
;
by county;
if cnt = . then cnt = 0;
if 0 <= cnt <= 4 then cntrng= 1;
else if 5 <= cnt <= 10 then cntrng=2;
else if 11 <= cnt <= 20 then cntrng=3;
else if 21 <= cnt <= 30 then cntrng=4;
else if 31 <= cnt then cntrng=5;
xtext = cats(name)||' ('||cats(cnt)||')';
format cntrng genrng.;
```

```
run;
```

```
proc gmap
data=male
map=MDCounties;
id county;
choro cntrng ;
format cntrng genrng.;
run;
```

HERE IS THE OUTPUT...

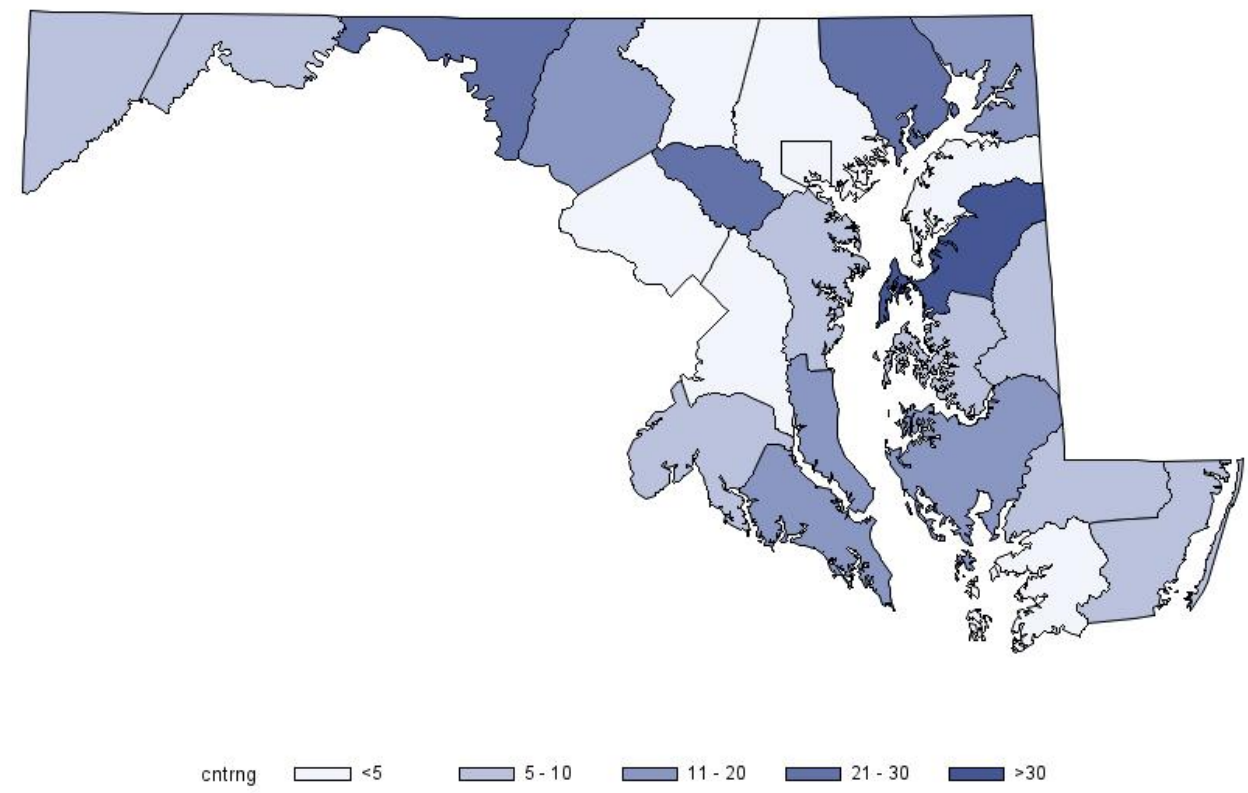


Figure 5. Map plotting the Male response ranges

ANNOTATE THE MAP

Like a silly Sith Lord I thought I could annotate the state map in a similar way that I had on previously programmed maps of the United States of America. That turned into a huge complicated nightmare. I kept either getting weird error messages or the map would run without any error messages but the annotations would still be missing. Or the program would work just as I wanted it to, but the solution was too overwhelming for a beginner to try to conquer. Finally, I stumbled across the solution of using a SAS macro called %MAPLABEL that will give you the annotations you are seeking.

First Step is to merge the dataset MDCounties created with the PROC GPROJECT with the Male dataset created earlier:

```
data MaleCounties;
merge MDCounties male;
by county;
run;
```

Then call the %ANNOMAC macro which tells SAS to have the annotate macros ready to be used:

```
%annomac;
```

The next step is calling the %MAPLABEL macro which will create the annotate dataset to be called in PROC GMAP in the annotate statement. MDCounties is the dataset created from PROC GPROJECT. MaleCounties is the dataset created in the previous step. MDAnnotate is the name the dataset with the annotation directions will be called. XTEXT is the variable we want to use as the labels for the counties. COUNTY is the variable describing how the dataset is sorted. Font has the name of the font we want to use. Color lets you choose the color the labels will be written. And Size describes how big the label should be when written.

```
%maplabel(MDCounties, MaleCounties, MDAnnotate, xtext, county,
font='Tahoma/bo', color=red, size=1.2);
```

Finally, the map is ready to be created with the ANNOTATE statement added:

```
proc gmap data=MaleCounties map=MaleCounties;
  id county;
  choro cntnrg / anno=mdannotate ;
run;
quit;
```

HERE IS THE OUTPUT...

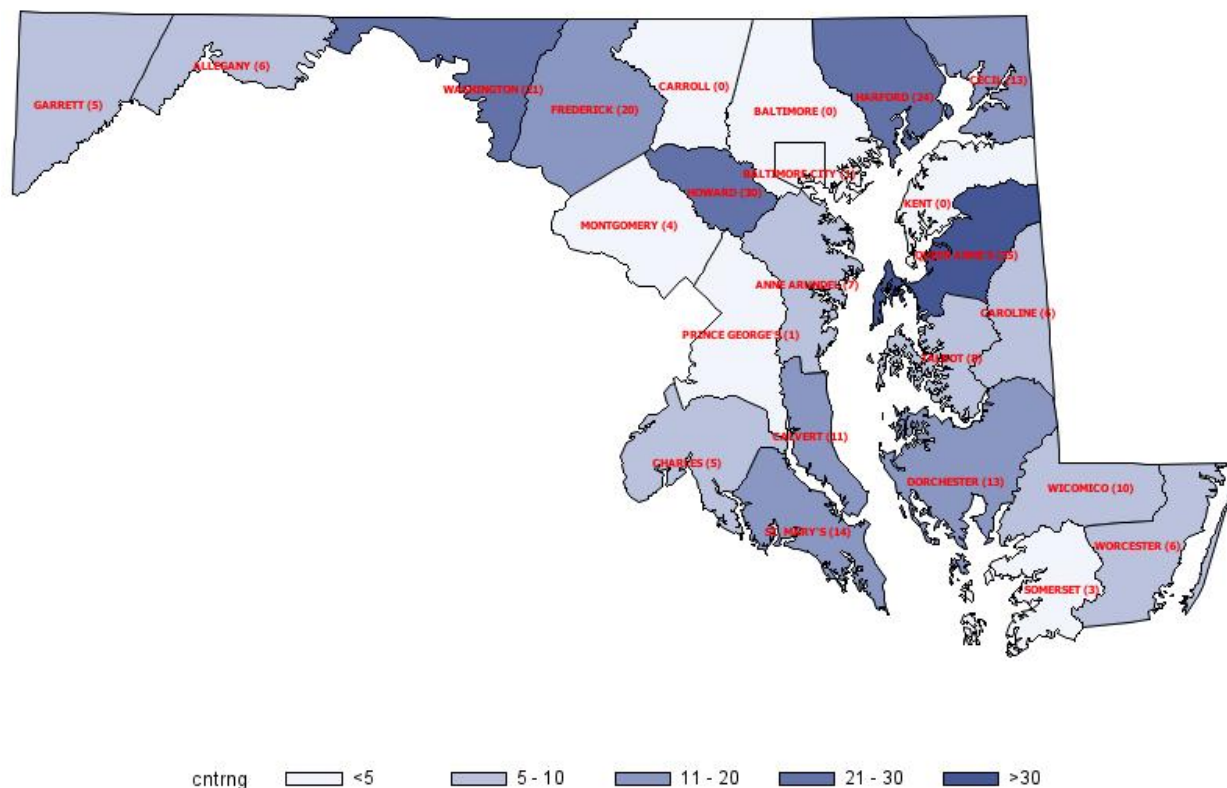


Figure 6. Map with the labels of the Male response numbers

ADDING A LEGEND

Legends help explain what the colors and numbers represent. In SAS there are a lot of options that can be used to create the legend. You can designate where on the map to place the legend, you can create labels for the values, you can add titles to the legend, and you can decide to display the values horizontally or vertically.

```

LEGEND1 LABEL=(HEIGHT=1 POSITION=TOP JUSTIFY=CENTER
"Ranges of the Number of Males Who Have Completed a Survey")
VALUE=(HEIGHT=1 "Less than 5" "5 to 10" "11 to 20"
"21 to 30" "More than 30")
ACROSS=1 DOWN=5 POSITION = (bottom outside left)FRAME
MODE=PROTECT ;

```

HERE IS THE OUTPUT...

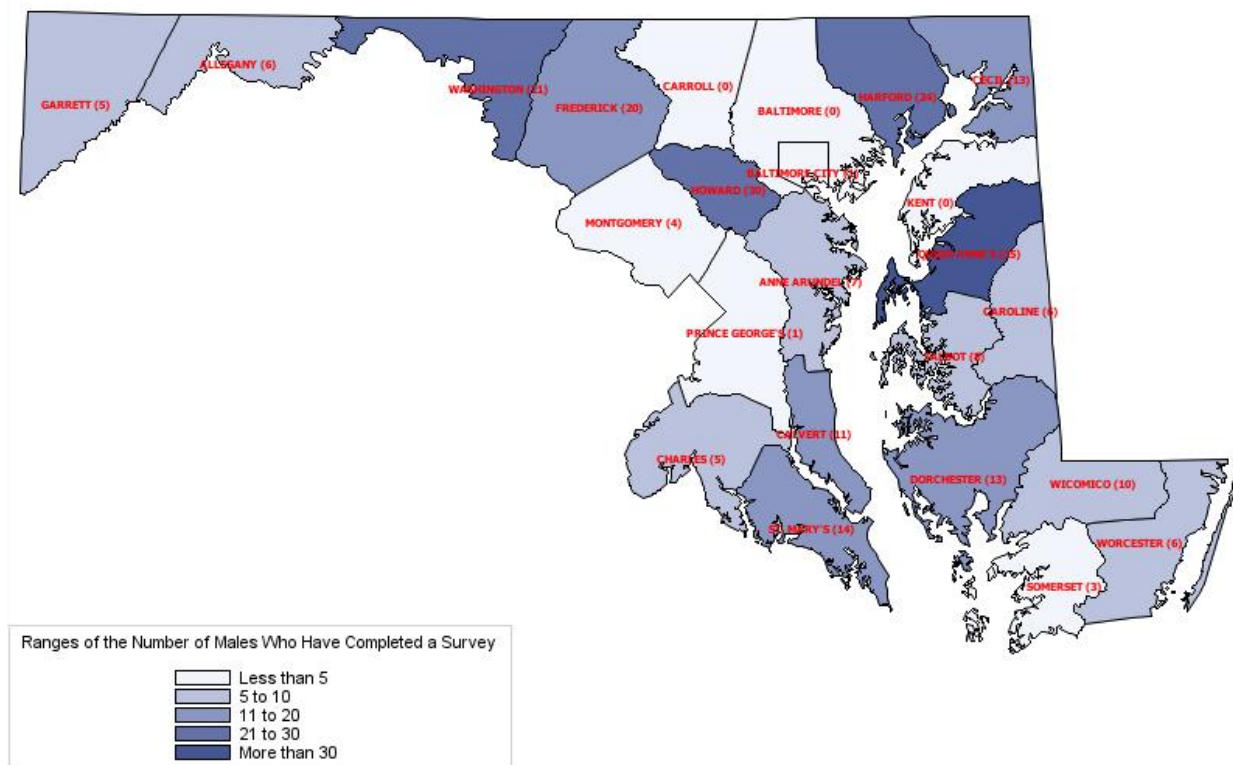


Figure 7. Map with the labels of the Male response numbers - added in a legend

SPECIFY COLORS AND PATTERNS

Sometimes you would prefer not to use the colors and patterns SAS uses as the default. You can specify your own patterns and colors using a PATTERN statement. Play around with your colors because on screen or when printed the color you think you requested might not be what you think you are seeing.

```

pattern1 v=ms c=white;
pattern2 v=ms c=yellow;
pattern3 v=ms c=magenta;
pattern4 v=ms c=mediumblue;
pattern5 v=ms c=purple;

```

HERE IS THE OUTPUT...

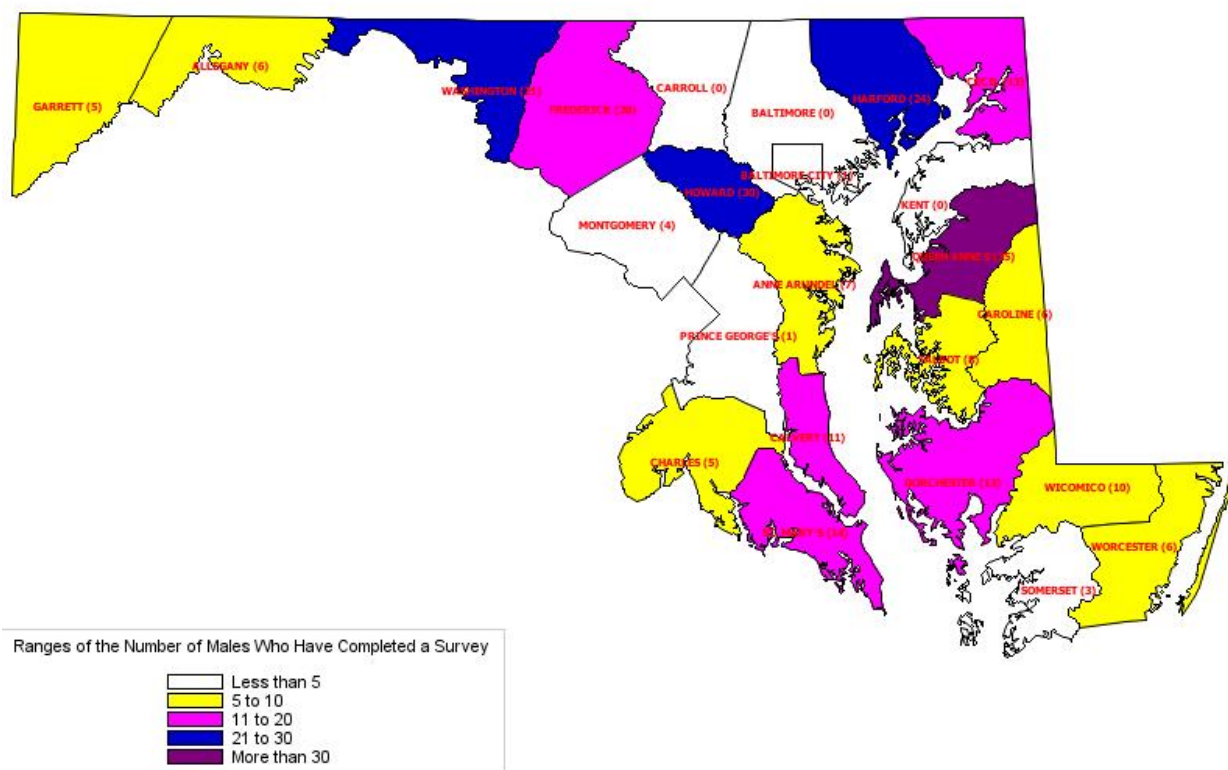


Figure 7. Map with the colors specified

You can also specify patterns, but I would suggest keeping these to a minimum for fear of confusing your audience. The V=MS creates a solid pattern. For the other patterns, the number at the end indicates the angle of the lines and the X in the middle indicates if it is a crossing pattern.

```
pattern1 v=M2N90 c=brown;
pattern2 v=M2X45 c=yellow;
pattern3 v=M3X0 c=magenta;
pattern4 v=M4N135 c=mediumblue;
pattern5 v=MS c=purple;
```

HERE IS THE OUTPUT...

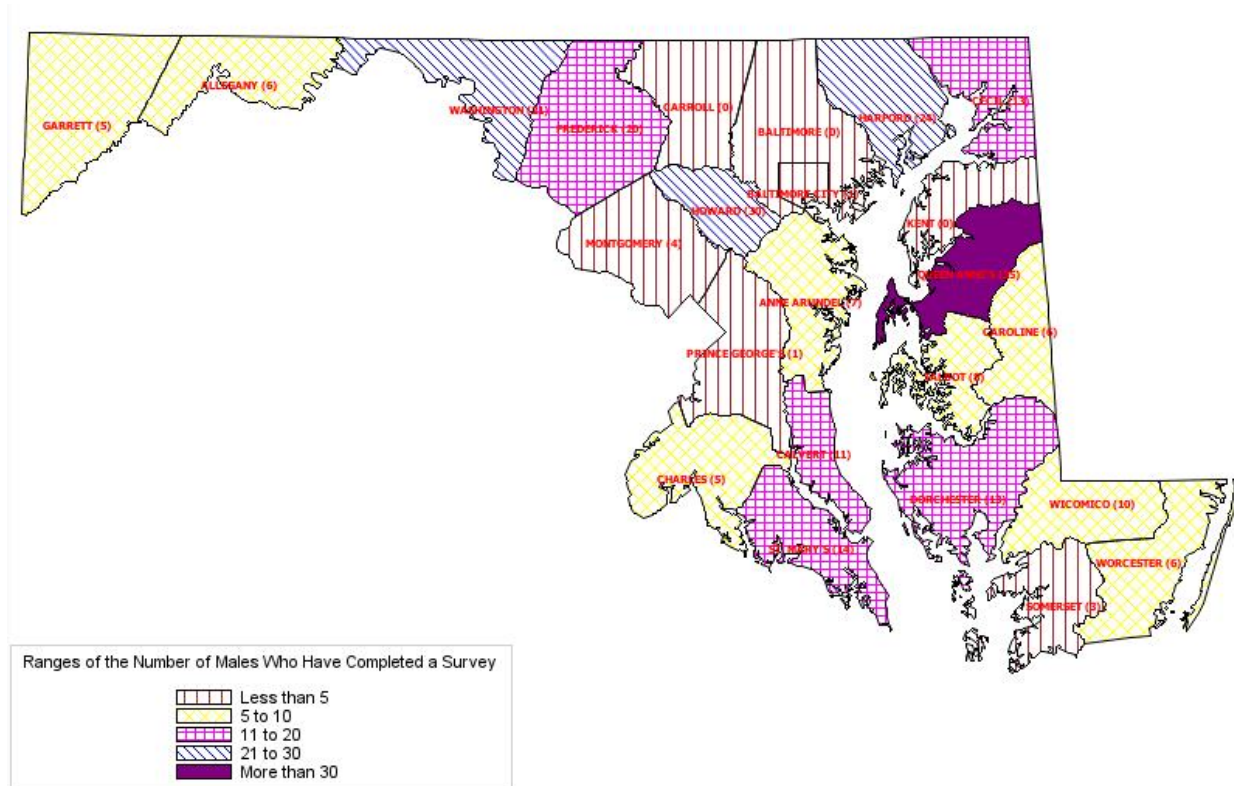


Figure 8. Map with the patterns and colors specified

TITLES VERSUS NOTES

Usually when we create maps we automatically use a title statement to label the output. However, when we use PROC GMAP we can instead use a note in the procedure to label the outputted map. The difference is in the real estate used. The title takes up a portion of the map display area which may cause problems with a map of a different shape or a bigger map causing issues with the legend display. Whereas, the note is a part of the map display and can help with oddly shaped maps, but be forewarned since they share space your note could overwrite your map. Notice in Figure 9 the amount of white space between the titles and the map and then look at the difference in Figure 10. You can also use both titles and notes if you have a lot of explanations you need to present.

Titles (with a footnote):

```
title1 "This is an Example of the Title Statement";
title2 "This is for the State of Maryland";
title3 "These are Males Who Have Completed the Survey";
footnote j=r "When will you see Star Wars 7?";
```

HERE IS THE OUTPUT...

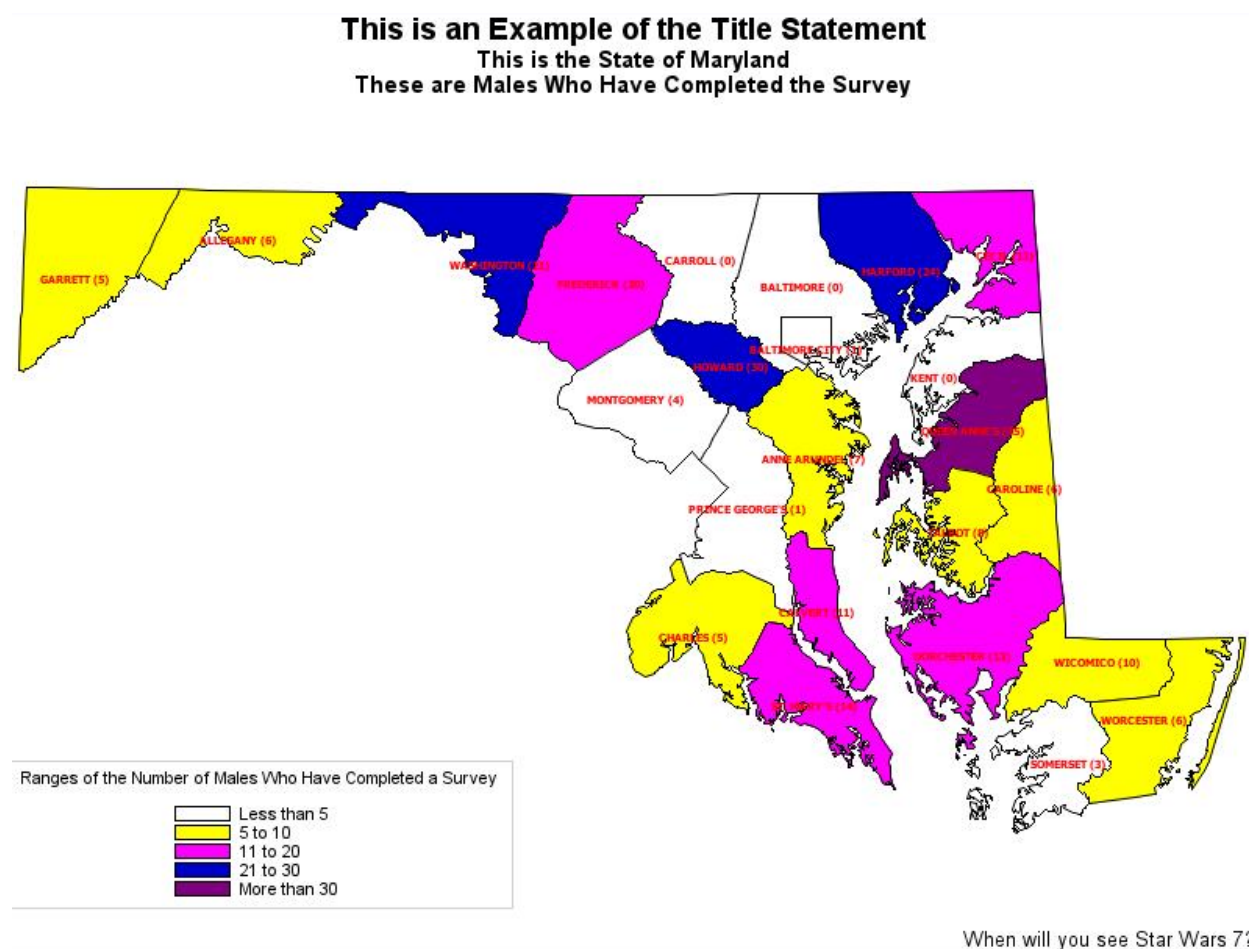


Figure 9. Title statements

Notes:

```
proc gmap data=maleCounties map=maleCounties;
  id county;
  choro cntnrg / anno=mdannotate legend=legend1;
  note 'This is an Example of the Title Statement '
    j=1 'This is for the State of Maryland '
    j=1 'These are Males Who Have Completed the Survey ';
run;
quit;
```

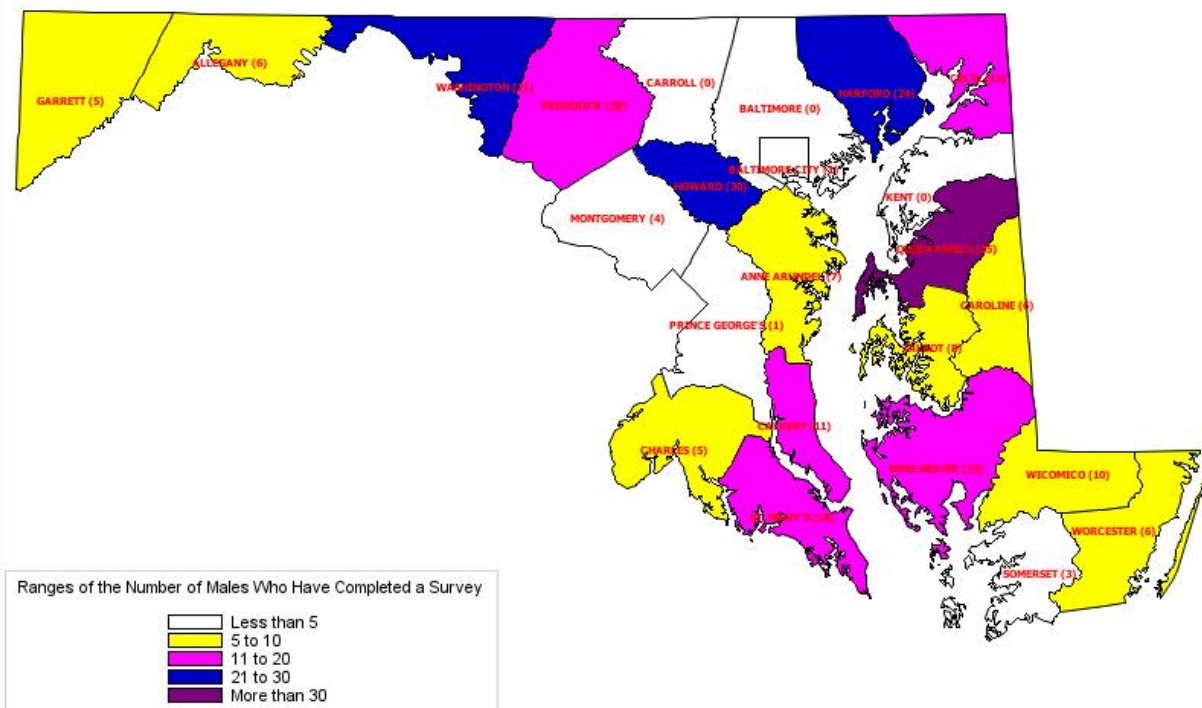


Figure 10. Note statements

```

title1 "This is a Survey of Devoted Star Wars Fans";
title2 "These are for the State of Maryland";
title3 "These are Males Who Have Completed the Survey";
footnote j=r "When will you see Star Wars 7?";

proc gmap data=maleCounties map=maleCounties;
  id county;
  choro cntnrg / anno=mdannotate legend=legend1;
  note 'Anyone else living with a Star Wars obsessed fan?'
    j=l 'Anyone else hoping the movie lives up to their fans expectations?'
    j=l 'Anyone else know more about Star Wars than they ever cared to
know?';
run;
quit;

```

HERE IS THE OUTPUT...

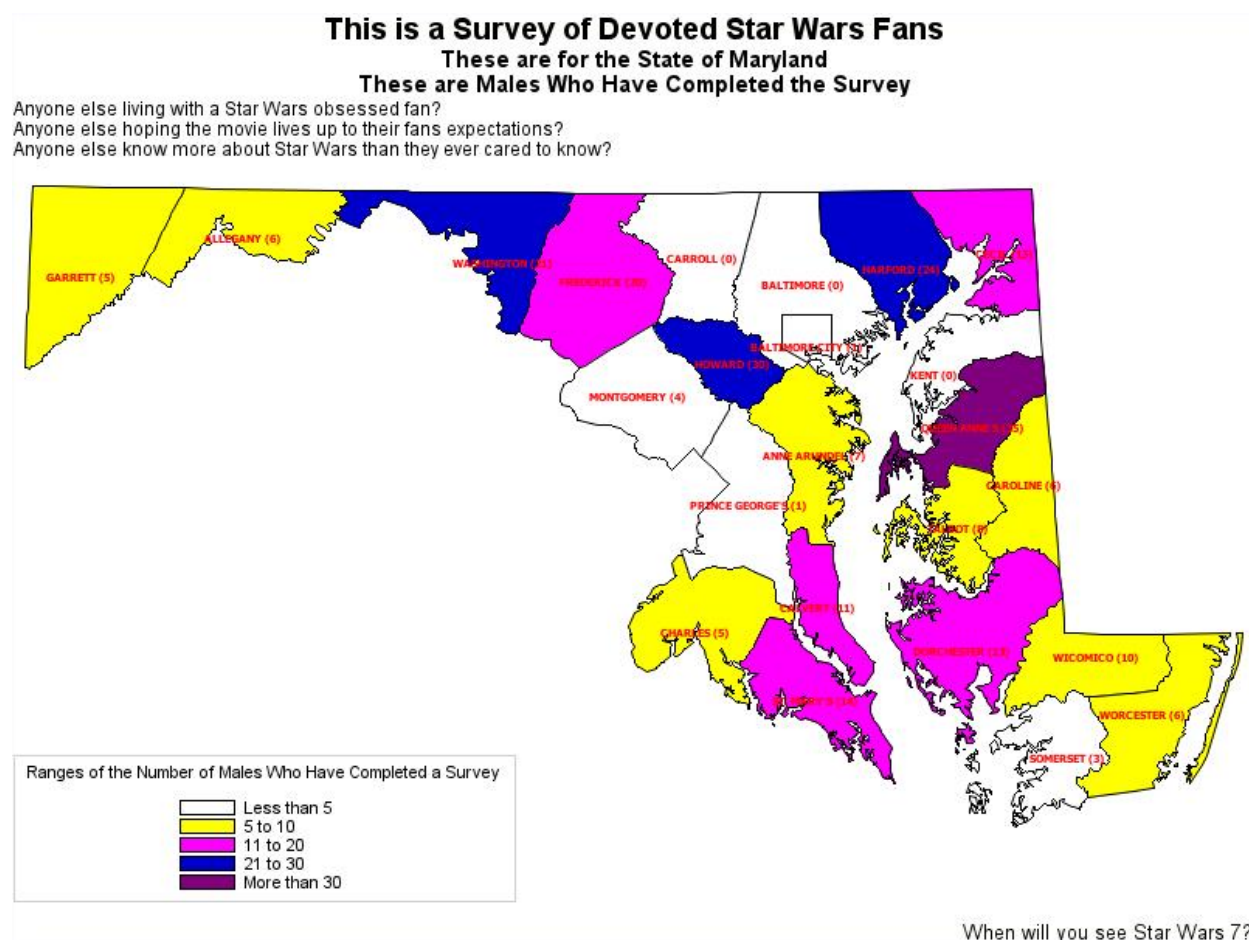


Figure 11. Both Title and Note statements

PROBLEMS WITH AUTOMATION

It is very crucial that you know and understand your data. With maps it might seem like a great idea to add in a macro to run your next map to display the females who have completed the study, but watch out if a category is missing. Notice on the legend you see 4 lines displayed with the 5th one missing. You might assume this means that no counties had greater than 30 responses. Unfortunately, once you look at the data you realize there are no counties in the 21-30 range and that Howard County falls into the More than 30 category. This occurs because the legend labels have no correlation to how SAS views the data. To resolve this you will need to set up several macros to first determine the number of categories and then select which legend to use based on the numbers needed.

HERE IS THE OUTPUT...

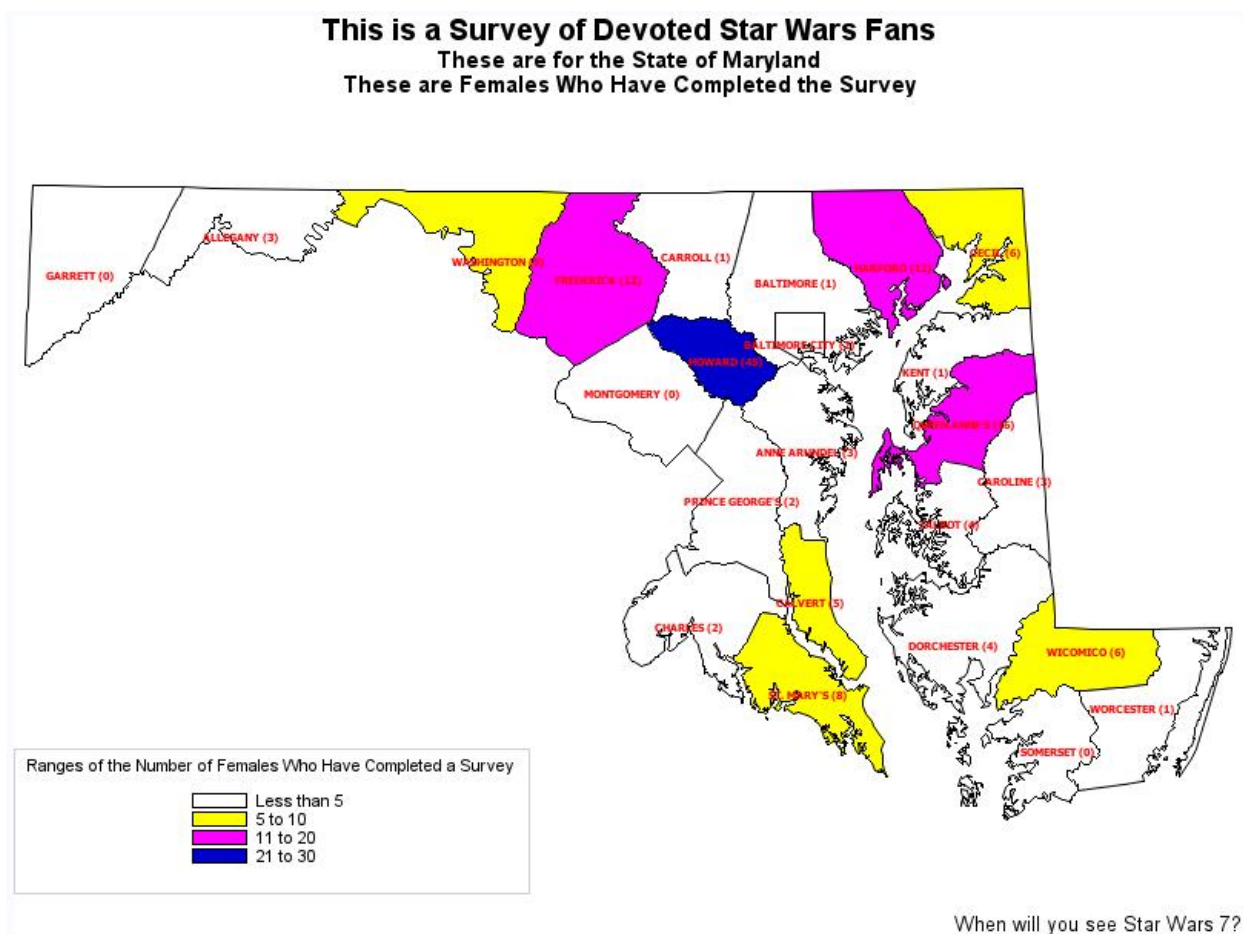


Figure 12. The Female categories are incorrect

CONCLUSION

This paper explored some basic steps to help the beginner SAS programmer get started on creating maps by counties for states. Once you have mastered these introductory steps there are several other SAS papers available online to help elevate your maps to the next level. Remember it may take several tries with multiple minor tweaks and adjustments to finally get your maps looking the way you want them to be displayed.

REFERENCES

Avrunin-Becker, S. "How to Make an Impressive Map of the United States with SAS/Graph for Beginners", SESUG 2014
 Zdeb, M. "The Basics of Map Creation with SAS/GRAPH", SUGI 29
 Eberhart, M. "Make the Map You Want with PROC GMAP and the Annotate Facility", NESUG 2008
 "The GPROJECT Procedure" Chapter 23 SAS document

ACKNOWLEDGMENTS

Many thanks to Michael Raithel, Rick Mitchell, and Sarah Woodruff.

CONTACT INFORMATION

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

Sharon Avrunin-Becker
 Westat
 1650 Research Blvd. – WB432

Rockville, MD 20850
Work Phone: (301) 738-8384
Email: SharonAvruninBecker@westat.com

DISCLAIMER

The contents of this paper are the work of the author and do not necessarily represent the opinions, recommendations, or practices of Westat.

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.