

Using SAS/GRAPH® to Generate State Metropolitan Area (MA) Maps

Francis J. Kelley
University Computing & Networking Services
The University of Georgia

ABSTRACT

This paper describes a SAS® macro which will generate all Metropolitan Areas (MA) in a state; it will include those portions of the MA which cross state boundaries as well. Several SAS/GRAPH OPTIONS (GOPTIONS) are set globally with macro variables, although defaults are provided wherever feasible. The data file used was TIGER/GICS™ (Topologically Integrated Geographic Encoding and Referencing/Geographic Identification Code Scheme) provided by the (US) Bureau of the Census; several other Census files provide similar information, and could be substituted.

necessary; the GICS file is ideal for this. Specifically, what was used was GICS9005.DBF. The "5" refers to the "GICS File type 5", which describes "Metropolitan Areas". These may be (briefly) described as one or more counties with a high degree of economic and social cohesion; population minimums for the "cores" of such places vary, but will fall in the range of 50,000 to 100,000. There are two basic types of MAs:

1. MSAs (Metropolitan Statistical Area's, formerly SMSAs or Standard Metropolitan Statistical Areas).
2. CMSAs (Consolidated MSAs).

CMSAs are in turn, composed of PMSAs (Primary MSAs). Census documentation describes MSAs as "...essentially freestanding MAs and not closely associated with other MAs"¹. CMSA's require a large urbanized area and a population of at least one million, as well as economic and social cohesion within both the PMSA and the larger CMSA. The file used, GICS9005, includes numerous records for all MAs, CMSAs, PMSAs, as well as records for those parts of each within a given state. A special code describes the "GICS Record type" for each record.

ORIGINS

This program was originally developed on an IBM® MVS® system, with the COSTAT2 (Census) file as input; MSA labels were added manually. The current program was started on the MVS system, then moved to an OS/2® system where it was significantly modified. Some features (notably, color TIFF files) required additional development using release 6.09 of the SAS System and another move, this time to an AIX® system. The data file used for this program was the Census Bureau's GICS file, as shipped on CD-ROM.

READING THE ORIGINAL DATA

The file provided by the Census Bureau, GICS9005.DBF is in a ".DBF" format, and may be read with PROC DBF on those versions which support it. In both OS/2 and Windows™, this requires SAS/ACCESS for PC

CENSUS FILES AND CENSUS GEOGRAPHY

The Bureau of the Census provides numerous data files, but for this program only a file containing the various codes and names was

File Formats. The data was read with this procedure, the following modifications were used:

1. SAS Data Set option DROP was used to remove all variables for which no use could be found.
2. RENAME was applied to the variable containing the name of the MA (ANPSADPI), to give the more descriptive AREANAME.

If other files are used here, they should have, at a minimum, the FIPS (Federal Information Processing Standard) codes for State, County and MA (MSA or CMSA). This program was written with the GICS file in mind, but STF (Summary Tape File) or COSTAT data could work as well. For labeling the MAs, a variable such as AREANAME is necessary. Most of the codes provided on the GICS file are of a character data type, even those that are numeric. This must be kept in mind, since it will be necessary to use these together with codes on SAS Map data sets; the map data set codes are of a numeric data type. The INPUT function was used to produce a numeric version of these character variables.

SETTING SOME OVERALL VALUES

Several macro variables are assigned at the start of the program. They may be easily altered, but they serve as useful defaults:

1. MAP is assigned the value COUNTY - this allows "MAPS.&MAP" to resolve as "MAPS.COUNTY", the COUNTY map file is much smaller than the COUNTIES map file, and its use is recommended in debugging.
2. FONT is assigned the value DUPLEX - this is used in the ANNOTATE data set. T1FONT and T2FONT are assigned the value F=&FONT. T1 and T2 refer to

TITLE and TITLE2 statements, T1FONT is the font to be used in TITLE; it includes the macro variable &FONT and will resolve to F=DUPLEX, if &FONT is DUPLEX. When using fonts that have a Bold form, for example SWISS, the assignment T1FONT= F=&FONT.B will result in the TITLE being assigned the SWISSB font. If a font is specified that does not exist, SAS/GRAPH will substitute SIMPLEX.

3. T1HT, T2HT, and ANNOHT set height for TITLE1, TITLE2, and the ANNOTATE data set, respectively. When routing to the monitor, heights should generally be smaller than when routing to a file.

THE DRAWMA MACRO

The DRAWMA macro accepts four keyword parameters:

IDFILE identifies the file with the Census information;

STATE names the state we wish to map (postal code abbreviation, GA, SC, NY, etc);

GRAFOPT names GOPTIONS we wish to specify

CANNO names the color we want for our TITLE and ANNOTATE data.

These last two may be left blank; the default output device in OS/2 is the monitor, and the default color is white.

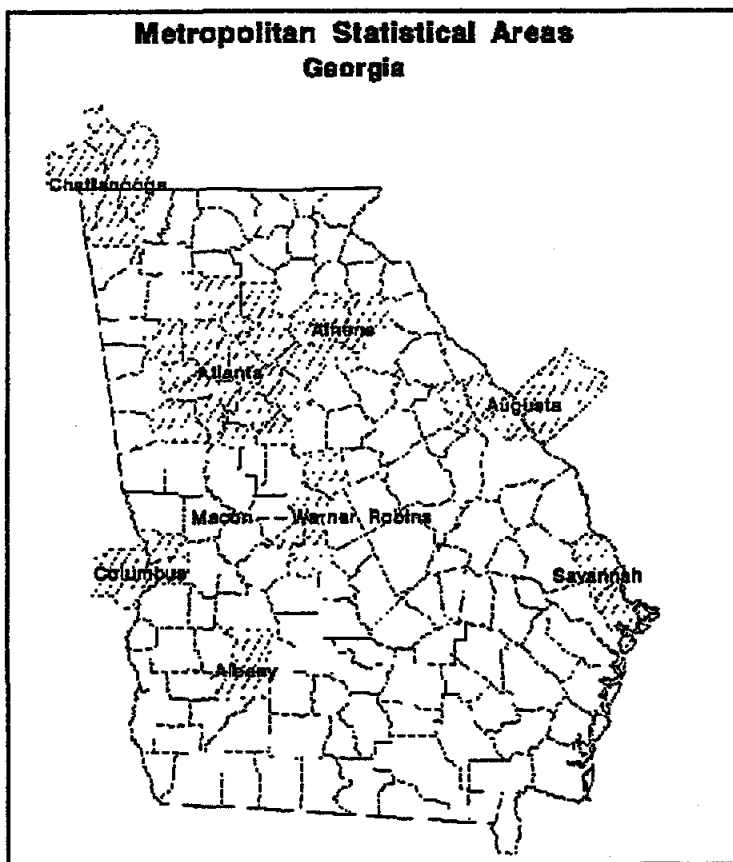
- a. The next step is to get a list of all the MA codes within the target state. The GICS record type (GICSRT) 300 is the code for MA's, but those MA's which cross state boundaries will not have a state (postal) code (STUSAB). Instead, we look for GICSRT of 310 in the state; record type 310 will have

that part of the MA within the target state.
This is the TEMP dataset.

The sections of code used to create the data sets POINTS and CENTERS, as well as the Annotate data set these two are used to create (CNTRPROJ) are taken from the SAS/GRAPH Sample Library program GMAPCTR.SAS and are used by permission of SAS Institute.

- b. MERGE this with the Census data set (and this time select for GICSRT 300) and we get the correct AREANAME. This is the MSALIST data set. In the course of creating this, a short message is written to the LOG if the MA is, in fact, a CMSA.

- c. A second data set, CNTYLIST, is created in much the same way, this time selecting on a GICSRT of either 329 or 331 (county in MSA or CMSA). CNTYLIST is now sorted by STATE and COUNTY, merged with the county map file we have selected, and the results written to MSAMAP. This will map all the counties in the MA's. MSAMAP is again sorted, this time by MSA STATE COUNTY, and PROC GREMOVE is used to get rid of all but the MA boundaries.



Calculating the centers of the MAs (to place the labels on the map) is a two-step process; first the number of points (observations) within each MA is calculated and written to POINTS, then, using a method described by Adrian Bowyer and John Woodward in "A Programmer's Geometry", the centers of the MA's are found and the labels written to the ANNOTATE data set. Following this, all the map files, as well as the ANNOTATE data set are concatenated into

one file, ALLMAPS and converted, using PROC GPROJECT, into a standard map format. The map and ANNOTATE files are then separated into MAPPROJ and CNTRPROJ. MAPPROJ is the map file and is used in PROC GMAP as both the map and the data file. CNTRPROJ is the ANNOTATE file. The variable ORD is used to set an "order" for the maps (and colors). ORD is specified on the ID statement together with MSA and COUNTY (it seems the "pen" is not

- d. To get the AREANAME variable back, MSAMAP is merged with MSALIST. A map of all counties both in the state and MA is created, as well as a map of the state (from MAPS.STATES).

lifted if you do not specify something like this). As a final step, the data sets created are deleted using PROC DATASETS. On systems where space is a problem, this should be done as soon as is practical - for example, once ALLMAPS has been created, no other file is needed.

ADDITIONAL COMMENTS AND OBSERVATIONS

At present, this macro requires more knowledge of SAS/GRAPH than many users may have; Certainly it could be made more user-friendly, although the price for that will be increased processing time. When creating a file, more effort is required than seems necessary, this could be changed. The labelling provided by the ANNOTATE data set is generally adequate, but sometimes the labels can end up on top of each other; hand-coding this may be the only solution for some states.

GETUSGEO.SAS

```
filename dbfile
                'a:gics9005.dbf' ;
libname perm XPORT
                'c:\sesug\usgeo.xpt' ;

proc dbf db3=dbfile out=perm.usgeo
  (drop=chariter logrecpn partrec anrc
  aianace aianafp aianacc artli blk blkgr
  tractbna congdis concitce concitfp concitcc
  concitsc cousubce cousubfp cousubcc
  cousubsc extcitin intuc placece placefp
  placecc placedc placesc urbarea uasc
  sac1-sac10 arealand areawat pop100 spflag
  ctbnafg wtrinlnd wtrcoast wtrgtlk wtrterr
  cosubfp2 cosubcc2 placefp2 placecc2
  namecor2 rename=
  (ansadpi=areaname) );
run;
```

DRAWMA.SAS

```
options nomprint nomacrogen symbolgen;
*
* Set some overall values for the program
*
%let map = county;
* use COUNTY (small) or COUNTIES (big) ;
%let font = duplex;
* select font for labeling the MSAs
;

%let t1font = f=&font;
* TITLE1 font - ".B" selects "BOLD"
;
%let t2font = f=&font;
* TITLE2 font - if bold n/a, you get
;
* - SIMPLEX, so be sure
;
%let t1ht = h=1;
* Title1 height - when printing, use a
;
%let t2ht = h=.75;
* Title2 height - larger height than
;
%let annoht = .75;
* MSA Label ht - when displaying
;

%macro drawma
  (idfile=,state=,graftopt=default, canno=white);

  %let state = %upcase(&state);
  %if %upcase("&graftopt") ^= 'DEFAULT' %then
    %do;
    %let gopt = goptions &graftopt ctitle=&canno
    %str(;);
  %end;
  %else gopt =;
  &gopt;
  *
  * Once the full data file has been extracted,
  * it must be reduced to two subsets:
  * 1 - A list of all MSAs and their codes in the;
  * target state. Call this MSALIST
  * 2 - A list of FIPS codes for all counties
  * within the MSAs. Because MSAs
  * may cross state lines, these codes
  * may be in states other than the target
  * state. Call this CNTYLIST.
  *
  * The variable GICSRT contains the codes
  * for the GICS record types. Selecting '310'
  * will get you all the information for that
  * portion of the MSA within a given state.
  * We can then use this to get the complete
  * MSA list (including the name).
  *
```

```

data temp ;
  set &idfile end=last;
  msa = input(msacmsa,4.) ;
  keep msacmsa msa;
  if stusab = "&state" and gicsrt = '310'
    then output;
  if last then call
symput('statnam',trim(stname("&state")));

data msalist;
  merge temp(in=a) &idfile;
  by msacmsa;
  retain comma ',' ;
  keep msa areaname stusab ;
  if (a = 1 and gicsrt = '300') then do ;
    loccomma = index(areaname,comma);
    areaname = substr(areaname,1,(loccomma - 1));
    * If this is really a CMSA, write ;
    * a message to the LOG. ;
  if cmsa ^= '99' then do ;
  put

  *****/
  'NOTE --' /
  'The Metropolitan Area: ' /
  areaname $45. /
  'FIPS Code: ' msa 4. /
  'is a Consolidated Metropolitan Statistical /
  Area (CMSA)' /
  *****/;

  end;
  output;
end;

data cntylist;
  merge temp(in=b) &idfile ;
  by msacmsa ;
  state = input(statefp,2.) ;
  county = input(cnty,3.) ;
  keep msa state county areaname stusab gicsrt
  cmsa pmsa;
  if (b = 1 and gicsrt in('329' '331')) then output;

proc print data=msalist ;
title "MSA List" ;
title2 "Selected State: &statnam" ;
run;

proc print data=cntylist ;
title "County List";
title2 "Selected State: &statnam" ;
run;

proc sort data=cntylist ;
by state county ;

```

```

data msamap;
  * Get every county in the MSA. ;
  merge cntylist(in=c) maps.&map;
  by state county;
  if c;

proc sort data=msamap;
  by msa state county ;

  * ;
  * We need a map file with the map info ;
  * for the MSAs we want. Use GREMOVE ;
  * to get rid of the STATE and COUNTY ;
  * stuff. We can add dummy values back ;
  * later. ;
  * ;

proc gremove data=msamap out=msamap ;
  by msa;
  id state county ;
run;

  * ;
  * As a result of PROC GREMOVE, the map ;
  * of our MSAs does not have the variable ;
  * AREANAME anymore. We can get it ;
  * back by merging MSAMAP with ;
  * MSALIST. ;
  * ;

data msamap;
  merge msamap msalist ;
  by msa;

  * ;
  * Now set up the maps for county and state ;
  * ;

data cntymap;
  merge cntylist(in=d) maps.&map ;
  by state county ;
  ord = 1 ; * set the order ;
  msa = . ; * dummy value for MSA ;
  if (state = stfips("&state") or d=1) then output;

data statemap;
  set maps.states;
  ord = 3 ; * set the order ;
  county = . ;
  * dummy value for COUNTY ;

  msa = . ; * dummy value for MSA ;
  if state = stfips("&state");

```

```
*****;
* The sections of code used to create the ;
* datasets POINTS and CENTERS, as well as ;
* the Annotate dataset these two are used to ;
* create (CNTRPROJ) are taken from the ;
* SAS/Graph Sample Library program: ;
* GMAPCTR.SAS. This program was ;
* developed at SAS Institute, and is with ;
* provided with SAS/Graph. It may be found ;
* will all current implementations of ;
* SAS/Graph. ;
*****;
```

```
data msamap(drop=npoints) points(keep=x y npoints
  rename=(x=xlast y=ylast));
  set msamap;
  by msa notsorted;
  retain flag 'M';
  ord = 2 ; * set the order ;
  state = . ;
  * dummy value for STATE ;
  county = . ; * dummy value for COUNTY ;
  output msamap;
  npoints + 1;
  if last.msa then do;
    output points;
    npoints=0;
  end;
run;
```

```
data centers(keep=state county flag ord msa style
  xsys ysys when function text x y color size);
  length function color $8 text $35;
  retain savptr 1 xold yold 0 when 'a' xsys ysys '2'
    function 'label' size &annoht color "&canno"
    flag 'C';
  retain style "&font";

  * specify a font (comment out if not needed. ;
SET points;
  * Repeat for each county ;
  xcg=0; ycg=0;
  * xcg,ycg = centers of gravity ;
  aresum=0;
  * aresum = area of polygon ;
  firstpnt=1;
  * Flag means first county point ;
  endptr=savptr + npoints - 1;
  * Set ending pointer for MSAMAP. ;
  do ptrm=savptr to endptr;
  * Repeat for each county point ;
  SET msamap(drop=flag) point=ptrm nobs=nobsm;
  if firstpnt then do;
  * If it is first county point: ;
  xold=x; yold=y;
  * Save first county point ;
```

```
savptr=ptrm + npoints;
  * Set next starting pointer ;
  firstpnt=0;
  * Reset flag ;
end;

  * ;
  * xold,yold = previous point in polygon ;
  * x, y = current point in polygon ;
  * aretri = area of triangle formed by last, ;
  * previous, and current points ;
  * ;
```

```
aretri=(xlast - x)
  * (yold - ylast) + (xold - xlast) ;
  * (y - ylast) ;
  xcg + aretri ;

  * ;
  * (x + xold) ;
  * Update XCG and YCG for ycg + aretri ;
  * (y + yold) ;
  * each county point aresum + aretri ;
  * Accumulate total area of polygon by ;
  * summing triangle areas (Aresum = double ;
  * the true area) ;
  * ;
```

```
xold=x; yold=y;
  * Save previous point ;
end;
```

```
areinv=1.0 / aresum;
  * Find inverse of total area ;
  x=((xcg * areinv + xlast) * 0.333333);
  * Find centers ;
  y=((ycg * areinv + ylast) * 0.333333);
  * of gravity ;
  text=areaname;
  * Text=annotate variable ;
  ord = 0 ;
  * Assign an ORDER value ;
  output;
  * Output the observation ;
run;
```

```
  * ;
  * Concatenate all MAP and ANNOTATE ;
  * datasets: MSAMAP, CNTYMAP, ;
  * STATEMAP and CENTERS. We want to ;
  * GPROJECT them at the same time - we ;
  * will separate them in a subsequent step. ;
  * ;
```

```
data allmaps;
  set centers cntymap msamap statemap;
  by ord msa state county;
run;
```

```
proc gproject data=allmaps out=mapfile;
  id ord msa state county ;
```

```
data mapproj cntproj;
  * Separate the points into
set mapfile;
  * two data sets again
if ord > 0 then output mapproj;
  * Map data set
else output cntproj;
  * Annotate data set
run;
```

```
title &t1h &t1font
  "Metropolitan Statistical Areas";
title2 &t2ht &t2font "&statnam";
```

```
proc gmap
  map=mapproj data=mapproj anno=cntproj;
```

```
id ord msa county ;
choro ord / discrete nolegend ;
```

```
pattern1 c=red v=E;
pattern2 c=blue v=M3N60;
pattern3 c=black v=E;
```

```
run;
quit;
```

```
proc datasets library=work;
  delete temp msalist cntylist
  msamap cntymap statemap
  points centers
  allmaps mapfile
  mapproj cntproj;
quit;
```

```
goptions reset=all;
run;
```

```
%mend drawma;
```

```
filename grafps 'c:\graphics.ps';
filename graftif 'c:\graphici.tif';
```

```
%drawma(idfile=perm.usgeo,state=ga,
  grafopt= device=os2 border,
  canno=white );
```

```
%drawma(idfile=perm.usgeo,state=ga,
  grafopt= device=tiffbii
  gsfname=graftif
  gsfmode=replace
  cback=white
  border,canno=black);
```

```
%drawma(idfile=usgeo,state=ga,
  grafopt= device=ps300
  gsfname=grafps gsfmode=replace
  cback=white
  border,canno=black );
```

```
run;
```

REFERENCES

TIGER/GICS™ (Topologically Integrated Geographic Encoding and Referencing/Geographic Identification Code Scheme) Census File, 1990 [machine-readable data file] / prepared by the bureau of the Census. Washington: The Bureau (producer and distributor), 1993.

This describes the data file itself, but the documentation is not so well described; it is provided on the same CD-ROM as the data, "Appendix A" (APDX_A.ASC) provides the descriptions of Census geography. Other essential files are the description of GICS record types (GICS_RT.ASC) and the actual data dictionary itself (GICS_DD.ASC).

¹ Appendix A of GICS documentation, "Census Geography".

SAS, SAS/GRAPH, and SAS/ACCESS software are registered trademarks or trademarks of SAS Institute Inc in the USA and other countries. IBM, MVS, OS/2, and AIX are registered trademarks or trademarks of International Business Machines Corporation. Windows is a registered trademark of Microsoft Corporation.® indicates USA registration.