Visualizing Census Data

Xanthippe Stevens and Larry Hoyle
Policy Research Institute
The University of Kansas

Abstract

At the Policy Research Institute we use a variety of SAS® techniques to create graphical displays of Census data. These techniques include PROC GPLOT, PROC GMAP (choro and prism), PROC GSLIDE, DATA Step Graphics Interface, and DATA Step writing HTML. We also use a variety of graphics device drivers and ODS. This paper displays a number of example graphics and associated SAS code.

Racial Data with PROC GPLOT

The 2000 Census allowed people to select one or more racial categories. In order to plot these data with historical data, we used the range of people selecting one race alone and those people plus those who reported that race in combination with any other race.

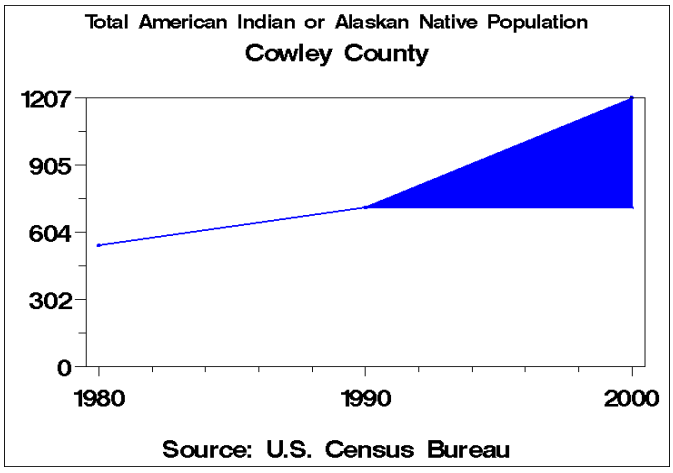

Figure 1 – Graph of American Indian or Alaskan Native Population in Cowley County, Kansas.

To produce these graphs in SAS, we used macro variables to record the maximum value and the maximum value divided by four to be used to scale the graph.

```
proc sql;
 select max(v_value), max(v_value)/4
 into :maxval, :scale from race00;
quit;
```

These variables are used in the axis2 statement to customize the y-axis. In order to create the pie shaped wedge, we used the overlay option in the plot statement. This plotted the area below the high number in the range with blue (pattern2) and below the low number with white (pattern1).

```
axis1 order=(1980 to 2000 by 10)
   label=none
   major=(height=2)
   minor=(number=4 height=1)
   offset=(2,2);

axis2 order=(0 to &maxval. by &scale.)
   label=none
   major=(height=1.5) offset=(0,0)
   minor=(number=1 height=1);

pattern1 color=white value=msolid;
pattern2 color=blue  value=msolid;

proc gplot data=hilo;
  plot lo*v_year hi*v_year / overlay
       areas=2
       haxis=axis1
       vaxis=axis2;
run;
```

Population Data with PROC GMAP

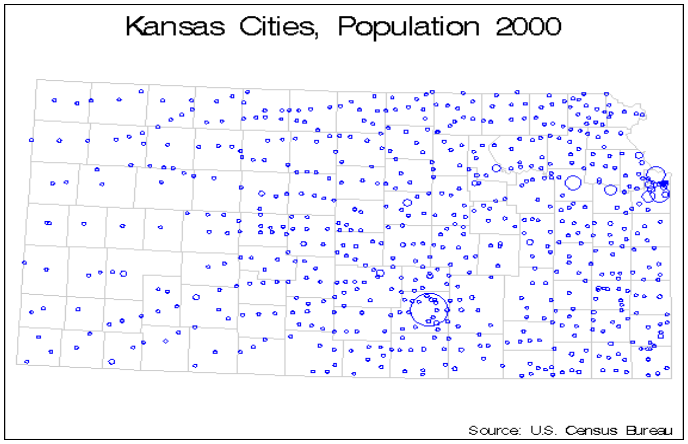

Figure 2 – Population of Kansas Cities

We created maps to show the 2000 population of Kansas cities. We used the annotate function to draw a circle at the location of each city over a map outlining county boundaries. We used the U.S. county and U.S. city map sets in the SAS maps directory for the county boundaries and center point for each of the cities. The diameter of each circle was calculated based on the city population.

```
data anno;
/*this routine annotates the city data
set to create pie shapes where the
radius is based on the 2000 population
*/
  length function style color $8;
  set city (keep=x y city value);
  xsys='2';
  ysys='2';
  hsys="1";
  style='pempty';
  position='5';
  color='blue';
  x=x;
  y=y;
  function='pie';
  angle = 0;
  rotate= 360;
  valrange=&maxval-&minval;
/* note: the variable "value" contains
the 2000 Population*/
  size = .5 + (value/valrange)*5;
  line=0;
  when='A';
  output;
run;
```

```
/* White background, black text, and a
light gray border for counties*/

goptions reset=global gunit=pct
  border cback=white device=gif570
  gsfname=giffile ctext=black
  ftext=swiss htitle=6 htext=3;

title1 'Kansas Cities, Population
2000';
footnote1 j=r 'Source: U.S. Census
Bureau ';

proc gmap map=dataloc.countymap
data=blanks;
  id county;
  choro cfill/discrete
              nolegend
              coutline=cxCCCCCC
              annotate=anno;
run;
quit;
```

For a more dramatic effect, we generated the same map using a black background and white fill in the pie shapes to lend a “night” appearance to the map.

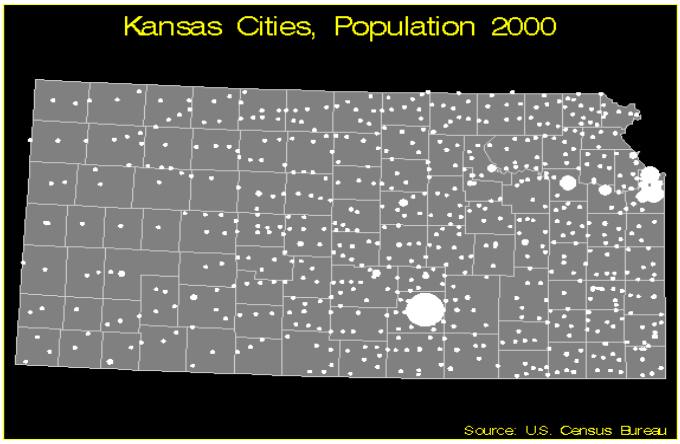

Figure 3 – “Night” view of Kansas Cities

Redistricting Data with ACTIVEX

We were asked to make a PowerPoint presentation on Congressional Districts in Kansas. This presentation was to inform the public and the legislative committee on redistricting at town hall meetings.

Figure 4 – Default view of prism map

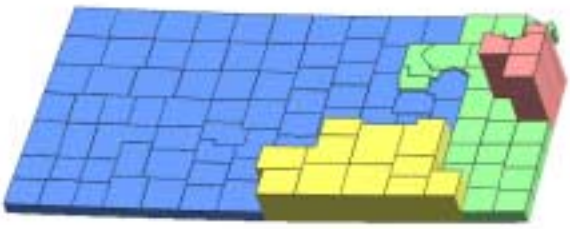

Figure 5 – Rotated view of prism map

We created prism maps using PROC GMAP to display data by district and output the maps with the activex device. This output file was an html file that contained, in this case, javascript that references an activex control. In the output file you can rotate the map, change the color scheme, add labels or a legend, or change text features. These changes, however, are not permanent. The specifications made in the goptions statement are the default options that will appear each time the html file is opened. For our presentation, I rotated the map and saved it as a .jpg to preserve the desired viewpoint. Figure 4 is the 2000 population by Congressional District in Kansas in the default view. Figure 5 shows the same map only rotated to clearly show the differences between the districts. We created similar maps showing racial and socio-economic data. For convenience in making multiple maps I used a macro variable (&varname) for the name of the variable from our data archive to use in both the map and as the filename.

```
data mapset;
      set maps.uscounty;
            where state=20;
      length fips 8;
      fips=county+20000;
run;
```

The map dataset used in this example is a subset of the county boundaries for the U.S. that is in the SAS map directory.

```
goptions reset=global device=activex
transparency;

ods html body="&varname..html"
path=ODSOUT;

proc gmap map=mapset data=ksdata;
  id fips;
  prism &varname./discrete
                  nolegend
                  coutline=black;
run;
quit;

ods html close;
```

For a closer look at a particular Congressional District, I used the TIGER data from the Census Bureau. This data consists of individual points that had to be "chained" or put in sequence in order to draw the boundaries.

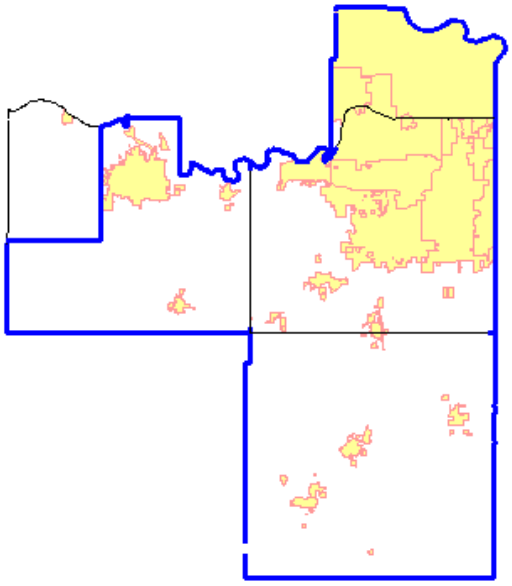

Figure 6 – Congressional District Three (Kansas)

This map outlines the county boundaries with a thin black line, the congressional boundary with a thick blue line, and all census-designated places in yellow. This map was created using PROC GSLIDE with an annotated data set containing a series of move/draw functions for

the county lines and poly/polycont for the city boundaries.

```
data annoplace;
/*annotate the city data*/
      length function color $8;
      set city (keep=x y line type tlid
      placer segment);
         by placer segment;

      if first.segment then do
            row=line;
            line=1;
            xsys = "2";
            ysys = "2";
            function="poly";
            when= 'A';
            color="cxFFFF99";
            style="msolid";
            output;
            end;
      else do
            row=line;
            function = "polycont";
            when= 'A';
            xsys = "2";
            ysys = "2";
            line = 1;
            style="msolid";
            color="cxFF9999";
            output;
            end;
run;

data anno;
/*concatenate annotated county and city
data*/
     set annocnty annoplace;
run;

goptions reset=all gunit=pct border
  device=gif gsfname=dist3map
  ftext=swissb htitle=6 htext=3;

proc gslide annotate=anno;
      run;
quit;
```

The key to drawing these maps is assigning the poly function to the starting point of each geographic unit and polycont function to each subsequent point in the unit. The code above shows the annotatation of the city units, the annotation of the county units is the same only the functions move and draw are used.

Using the same idea, I created a map showing the population in each township in this district to help visualize the population outside urban areas.

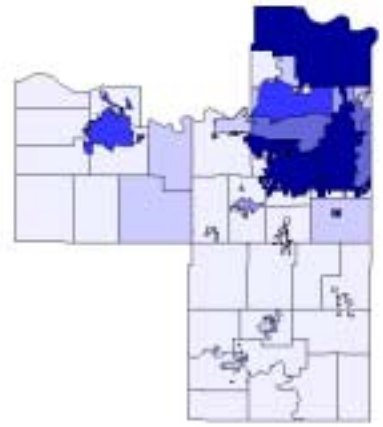

Figure 7 – Population by Township (Kansas Congressional District Three)

```
proc gmap map=mapset data=dist3;
      id cousubr;
      choro popvar/levels=6
                  coutline=black;
run;
```

Using the option levels=6 in the choro statement shades the townships into six levels based on the population. This makes it easier to see the differences in population without the need to define a color for each geographic unit.

Drill-Down Maps for the Internet

In order to create an easier interface to our data, we created a drill-down page for Kansas cities. This allows a user to select a county, view all of the cities within that county, and then upon selecting a city, view all of the reports and data that we have available for that city. The drill-down maps of Kansas and Kansas counties were created using SAS/GRAPH® with the java device driver.

Figure 8 – Map of Kansas

Figure 8 shows a java applet that displays the State of Kansas. A flyover is displayed as a user moves the mouse over a county. The flyover contains the id variable and choro variable from the PROC GMAP statement. In this case, the flyover contains the county name and county reference number as assigned by the U.S. Bureau of the Census.

Figure 9 – Map of the Cities in Douglas County, Kansas

The drill-down feature in these maps was created with the following code:

```
ods listing close;

ods html file="&geo..html"
  path=ODSOUT

  archive='http://www.ku.edu/pri/
   ksdata/sashttp/java/mapapp.jar'

  parameters=("DRILLDOWNMODE"="HTML")

  parameters=("DRILLPATTERN"=
    'http://www.ku.edu/cgiwrap/
    ippbrwww/city.pl?cityfips=
    {&txtplace}')

  parameters=("BACKCOLOR"="FFFFFF");
```

Followed by a DATA step with the latitude and longitude information and then completed with the following:

```
proc gmap map=temp data=temp;
      id idvar;
      choro txtplace/discrete
           nolegend
           coutline=black;
run;

ods html close;
ods listing;

quit;
```

This code was executed within a macro in order to create one page for each county in Kansas. This macro passes in the variable &geo used in the file statement. The drilldownmode parameter tells SAS to create html links with the pattern indicated in the drillpattern parameter.

The id variable, idvar, used in the gmap statement is a number that has a format that allows the name to display in the flyover.

It is important to note that in order to post these files on the internet and have them work, the mapapp.jar file must be on the server. The location of this file needs to be indicated in the archive statement.

Shaded Map with PROC GMAP

PROC GMAP will create shaded maps with equally spaced shading intervals automatically. We also use PROC GMAP to create maps with shading proportional to the displayed value. This map shows Kansas counties shaded in proportion to 1998 precipitation.

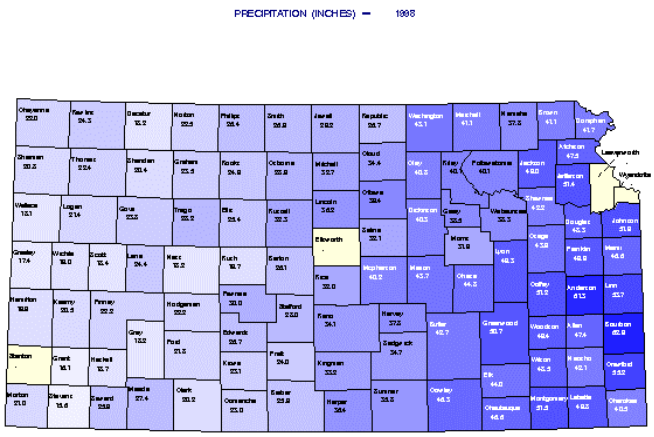

Figure 10 – Shaded Map of Kansas

The trick to doing this is to add a variable to the dataset with a shading value using CX*rrggbb* color values. A DATA Step can then write a

Goptions color= (……..)

statement, which can then be included back into the SAS program.

The shading values can also be used to write out an HTML table like this one which has shaded cells for each data value. This table can function as a fully detailed legend to the map. This table is sorted by county name. Optionally, the table could be sorted by shading value.

PRECIPITATION (INCHES) - 1998											
Allen	47.37	Doniphan	41.69	Jackson	48.98	Morris	31.87	Saline	32.07
Anderson	61.32	Douglas	48.34	Jefferson	51.39	Morton	20.99	Scott	18.41
Atchison	47.49	Edwards	26.69	Jewell	29.21	Nemaha	37.76	Sedgwick	34.66
Barber	25.86	Elk	43.99	Johnson	51.87	Neosho	42.13	Seward	25.85
Barton	26.12	Ellis	25.41	Kearny	20.46	Ness	18.16	Shawnee	42.16
Bourbon	62.9	Ellsworth	.	Kingman	33.23	Norton	22.47	Sheridan	20.37
Brown	41.06	Finney	22.19	Kiowa	23.14	Osage	43.9	Sherman	20.84
Butler	42.72	Ford	21.77	Labette	49.84	Osborne	28.86	Smith	26.85
Chase	44.8	Franklin	49.89	Lane	24.44	Ottawa	39.35	Stafford	27.99
Chautauqua	46.56	Geary	38.49	Leavenworth	.	Pawnee	29.97	Stanton	.
Cherokee	40.54	Gove	23.76	Lincoln	36.15	Phillips	26.36	Stevens	15.58
Cheyenne	22.02	Graham	23.48	Linn	53.67	Pottawatomie	40.12	Sumner	35.77
Clark	20.17	Grant	16.08	Logan	21.44	Pratt	24	Thomas	22.39
Clay	40.75	Gray	18.15	Lyon	49.29	Rawlins	24.27	Trego	28.22
Cloud	34.43	Greeley	17.37	McPherson	40.17	Reno	34.1	Wabaunsee	38.27
Coffey	51.24	Greenwood	50.7	Marion	43.71	Republic	26.7	Wallace	18.09
Comanche	23.01	Hamilton	19.87	Marshall	41.09	Rice	31.97	Washington	43.11
Cowley	46.28	Harper	36.38	Meade	27.42	Riley	40.08	Wichita	18.96
Crawford	55.16	Harvey	37.75	Miami	46.63	Rooks	24.88	Wilson	48.5
Decatur	18.22	Haskell	18.74	Mitchell	32.66	Rush	19.69	Woodson	49.35
Dickinson	40.33	Hodgeman	22.22	Montgomery	51.45	Russell	32.28	Wyandotte	.

Figure 11 – Table of shaded data values.

SAS code for producing the shaded table:

```
%macro gifhtm(srtby=county);
                                    /* write out a table of the data for GIFs */
  data _null_;
   file _webout;

   put "<P><CENTER>";
   put "<TABLE BORDER>";
   put "<TR><TD bgcolor=""#CCCCFF"" colspan=15
align=""center"">&TITLESTR";
   run;


                                /* print the rows */
  data _null_;
   set sextrct end=last;
   by rmod21 &srtby;
   file _webout;
                                /*  get the extrema for the plotted variable */
   retain xmax xmin;
   if _n_ = 1 then do;
     set minmax;
   end;
                  /* start a row */
   if first.rmod21 then do;
     put "<TR>";
   end;
```

```
                                /* shade the values like the map */
   ival=int(.499999 + &locolor +
      (value-xmin) * ( (&hicolor-&locolor)/(xmax-xmin) ));
   if ival=. then shade="&misshex" ;
   else
     shade =   put(&hicolor - ival + &locolor, hex2.) ||
           put(&hicolor - ival + &locolor, hex2.) ||
           "FF";
                                /* print the cells */
   put '<TD bgcolor="#'
                    shade +(-1)
                   '"> '
      '<TD><font  size=2>'                    county cntyfmt. '</font>'
      '<TD align="char" char="."><font size=2>' value best7. '</font>';
  run;

  data _null_;
   file _webout;
   put "</TABLE>";
   put "</CENTER><P>";
  run;
%mend gifhtm;
```

We also produce a color shaded map of population change with a long term trend graph of population inside each county. Techniques for creating these maps were described in an earlier MWSUG paper (Hoyle 1991).

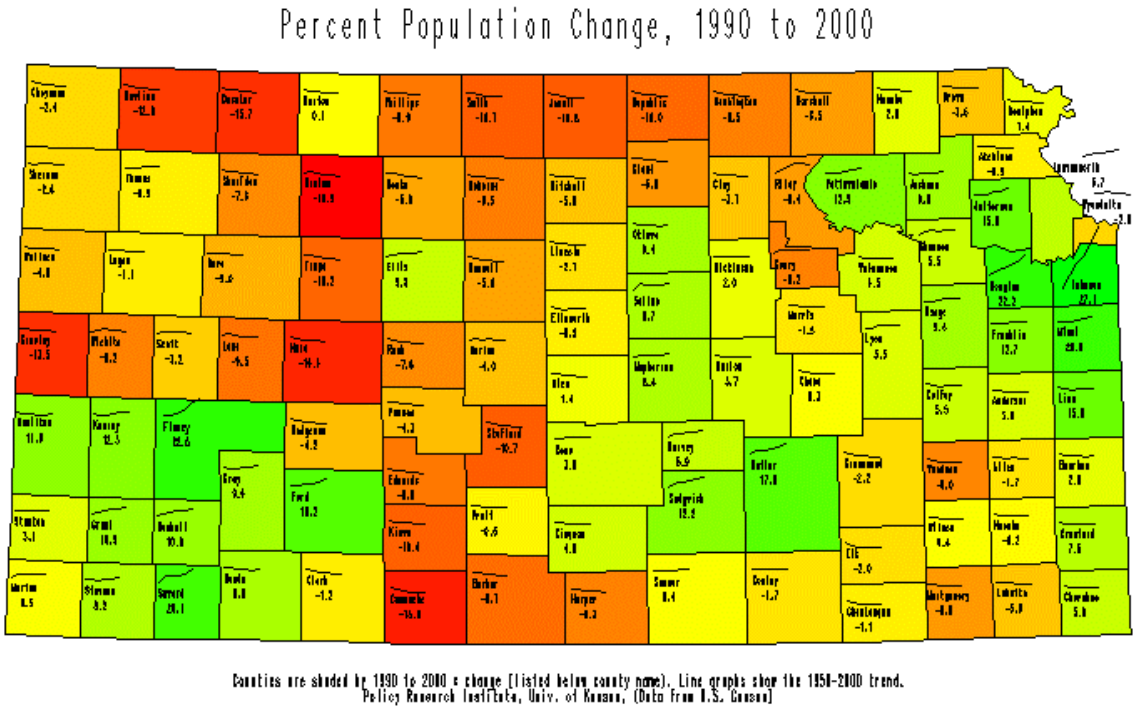

Figure 12 – Trend graph and shaded map of Kansas

Graphs using DATA Step Graphics Interface

The GIFANIM graphics driver allows for the creation of animated graphics in web pages. We use this driver to create animated population pyramids (see http://www.ku.edu/pri/ksdata/census/estpop/).

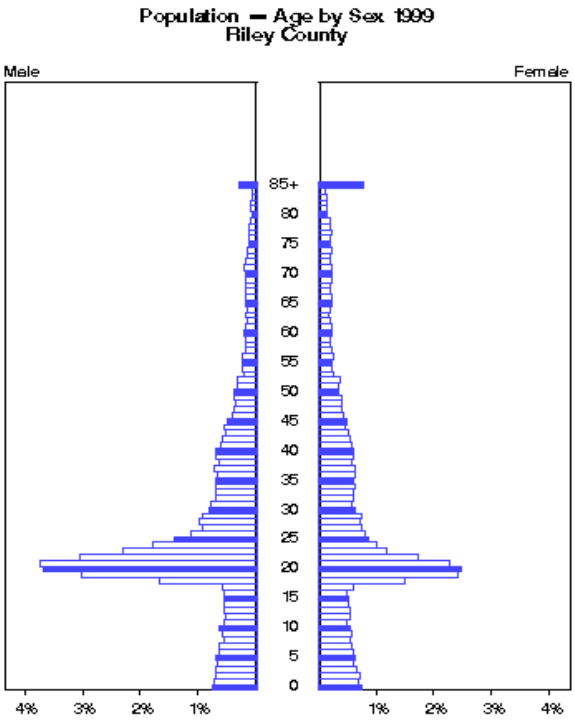

Figure 13 – Population Pyramid

The accompanying fragments of DATA Step Graphics Interface code show opening a new file:

```
data _null_;
  set withsum() end=last;
  by cnty year;

  where cnty<4;                          /* FOR DEBUGGING */

  length hrefg $ 100 hrefh $ 100 frefh $ 200;

  file LOG;
  if first.cnty then do;
    goptions  gsfmode=replace;
    hrefg='anPyr'||trim(cnty)||'.gif';
    hrefh='anPyr'||trim(cnty)||'.htm';
    frefh=trim("&outdir\"||left(hrefh));
    put hrefg=;
    put hrefh=;
    put frefh=;

          /* assign fileref for this county */

  rc=filename('PYROUT',trim("&outdir\"||left(hrefg)));
  sysmess=sysmsg();
  if rc then  put 'error assigning pyrout' rc= / sysmess;
  end;  /* first.cnty */
  else do;
    goptions gsfmode=append;
  end;   /* not first.cnty */
```

Drawing to the file

```
                                  /*  Create the graphic when the array is full */
if last.year then do;
  length ylbl $ 3;

                              /* initialize DSGI */
  rc=ginit();
  rc=gset('WINDOW',1,0,0,&winsiz,&winsiz);
  rc=gset('TRANSNO',1);
  rc=graph('clear','text');

                              /* assign colors */
  rc=gset('colrep',1,'CX4444FF');          /* regular bar  */
  rc=gset('colrep',2,'CX000000');          /* mult of 5 yr */
  rc=gset('colrep',3,'black');           /* text        */
  rc=gset('colrep',4,'black');           /* title*/
            /* title */
  rc=gset('texcolor',4);
  rc=gset('texfont',"&font");
  rc=gset('texheight',&ttlht);
  rc=gset('texalign','CENTER','BASE');
  rc=gdraw('text',&ttlx,&ttly+&ttlht,'1990 Percent Population by Age by Sex');
  rc=gdraw('text',&ttlx,&ttly,put(cnty,cntyfmt.));

  rc=gset('texheight',&mfht);
  rc=gset('texalign','LEFT','BASE');
  rc=gdraw('text',&leftmrg,&botmrg+(104*(&barwdth+&linspc)),'Male');

  rc=gset('texalign','RIGHT','BASE');
  rc=gdraw('text',&winsiz-&leftmrg,&botmrg+(104*(&barwdth+&linspc)),'Female');

          /* lines */
  rc=gset('linwidth',&barwdth);

                          /*  draw boxes */

  rc=gset('lincolor',3);
  rc=gdraw('line',2,&leftmrg,&leftmrg+&grwdth,&botmrg,&botmrg);

rc=gdraw('line',2,&leftmrg+&grwdth,&leftmrg+&grwdth,&botmrg,&botmrg+(103*(&
barwdth+&linspc)));

rc=gdraw('line',2,&leftmrg+&grwdth,&leftmrg,&botmrg+(103*(&barwdth+&linspc)),
&botmrg+(103*(&barwdth+&linspc)));

rc=gdraw('line',2,&leftmrg,&leftmrg,&botmrg+(103*(&barwdth+&linspc)),&botmrg);

  offset=&leftmrg+&cntrmrg+&grwdth;
  rc=gdraw('line',2,offset,offset+&grwdth,&botmrg,&botmrg);

rc=gdraw('line',2,offset+&grwdth,offset+&grwdth,&botmrg,&botmrg+(103*(&barwd
th+&linspc)));

rc=gdraw('line',2,offset+&grwdth,offset,&botmrg+(103*(&barwdth+&linspc)),&bot
mrg+(103*(&barwdth+&linspc)));
  rc=gdraw('line',2,offset,offset,&botmrg+(103*(&barwdth+&linspc)),&botmrg);
```

And finally, closing the frame and writing the proper terminator to the file.

```
rc=graph('update');
rc=gterm();
                              /* teminate the file after the last frame */
if last.cnty then do;
  data _null_;
    file pyrout recfm=n mod;
    put '3B'x;
  run;
end;
```

Examples in this paper and other examples of visualizing demographic data can be found on the Policy Research Institute web site at http://www.ku.edu/pri. Complete SAS code for the examples in this paper can be found at http://www.ku.edu/pri/ksdata/sashttp/mwsug01.

Contact Information:

Xanthippe Stevens
Policy Research Institute
The University of Kansas
1541 Lilac Lane, Suite 607
Lawrence, KS 66044-3177
(785) 864-9111
xan@ku.edu

Larry Hoyle
Policy Research Institute
The University of Kansas
1541 Lilac Lane, Suite 607
Lawrence, KS 66044-3177
(785) 864-9110
larryhoyle@ku.edu