

Paper GH-17

Using SAS® to Create Custom Healthcare Graphics

Barbara B. Okerson, WellPoint, Richmond, VA

ABSTRACT

SAS® graphics products can be used to create and deliver high-impact visuals, both as presentation-ready graphics and as data exploration displays. Using these graphics, users can explore, examine, and present data in an understandable manner while distributing their findings in a variety of formats to decision makers who can gain a quick, visual understanding of critical issues. Although SAS provides many ready-to-use graphics and graphic formats through SAS/GRAPH software and Output Delivery System (ODS) graphics, sometimes other graphics displays are needed. SAS provides the tools to create almost any desired graphic. This paper shows how SAS can be used to create graphics that are not directly available. Examples are from the healthcare industry.

Results included in this paper were created with SAS 9.1.3 or SAS 9.2 on a Windows XP platform, using Base SAS, SAS/STAT software, and SAS/GRAPH. SAS 9.1 or later is required for ODS graphics extensions.

INTRODUCTION

The effectiveness of a graph or graphic is determined by its ability to communicate information to its viewers. Graphs should increase understanding and represent the best way of displaying data, not just be aesthetically pleasing. The SAS system provides the ability to develop both basic and presentation-ready tools for data display. SAS/GRAPH software includes over 20 procedures that create maps, bar charts, plots, and other commonly used graphic data displays. ODS graphics add presentation quality graphics to many SAS/STAT and SAS/ETS procedures. SAS/IML also provides some ready-made graphic options. Additionally, SAS includes the Graph-N-Go application which allows the creation of standard SAS graphics without writing code. All of these graphics are customizable.

The sets of available and customizable graphics included within SAS products cannot possibly represent all graphics that might be used to display data and results. Rather than just creating customizable versions and enhancements of these provided graphics, the focus of this paper is using these graphics as a starting point to create new graphics.

Other papers have discussed the creation of custom graphics with SAS. These have emphasized the use of the graphics primitives, Graph Template Language, SG procedures (e.g. SGPLOT, SGPanel, SGSCATTER), or modifying style templates. The custom graphics in this paper use the SAS procedures in SAS/GRAPH as a base and use them in ways perhaps not intended, or at least not documented, to create graphics that are currently not part of the SAS graphics library. Custom graphics that are created and included as part of this paper are:

- Historical timelines
- Project timelines
- Waterfall charts
- Distance charts

Data sets used in this paper are either publicly available or are test data sets developed for the purpose of data modeling and technique demonstration and are not intended to represent the statistics of an actual population or account. The graphic techniques illustrated in this paper have been used in data presentations; in some cases they have been updated and/or enhanced for this paper. Partial SAS code is included in this paper; the selected code illustrates the techniques used to define the graphics. Complete code for all included graphics is available upon request.

EXAMPLE 1 – HISTORICAL TIMELINES

One of the most convenient ways to display a progression of events is with a timeline. This type of timeline is often referred to as a historical timeline and can be used to identify dates relating to any process or event. Historical timelines differ from the traditional data timelines – those that are used in longitudinal analysis as a means to show data change over time. For historical timelines, each time period represents an event time. Two examples of historic timelines are presented in this paper. In the first example, a historic style timeline is created that tracks event planning for a training session. The second example incorporates images into a more traditional historical timeline of medical history.

EXAMPLE 1a – EVENT TIMELINE

In the first timeline example, the X-Y axis is created with SAS/GRAPH Proc GPLOT, with the needle interpolation option used to mark the event points on the X-axis. The points are labeled with a SAS annotate data set. This example represents the basics only. To further enhance the usefulness of this type of timeline, different types of events can be identified by color, and a hierarchy of event importance can be created through font size.

**EXAMPLE 1a SELECTED SAS CODE**

```
/*Create Annotate Data Set*/
data anno;
length function color $ 8; retain when 'a' position '3' hsys '3' ysys
      '2' xsys '2';
set timeline;
if space > 0 then do;
  function='label'; size=3; style="'Calibri/it/bo'"; color='black';
  text=cat(' ',label) ;x=date-5; y=space; cborder= 'black'; cbox='beige';
output;end;
if space < 0 then do;
  function='label'; size=3; style="'Calibri/it/bo'"; color='black';
  text=cat(' ',label) ;x=date-5; y=space-2; cborder= 'black';
  cbox= 'beige';
output;end;
if space > 0 then do;
  function="symbol"; color='black'; style='marker'; text='C'; size=2;
  angle=0; x=date; y=space-.001; ;end;
if space < 0 then do;
  function="symbol"; color='black'; style='marker'; text='D'; size=2;
  angle=0; x=date; y=space+.32; end;
  cborder= ' '; cbox= ' ';
output; run;

/*Set Symbols and Axes*/
title2 c=black h=2.5 f='Calibri/bo' "Prevention Agency Training
      Schedule 2010";
axis1 order=(-14 to 14 by 1) major=none minor=none value=none style=0
      label=none;
```

```
axis2 label=none order=(15235 to 15305 by 10) minor=none;
symbol c=black i=needle;

/*Plot the Timeline*/
proc gplot data=timeline annotate=anno;
  plot space*date/vaxis=axis1 haxis=axis2;
  format date mmddyy5.;
run; quit;
```

EXAMPLE 1b – HEALTHCARE TIMELINE

This timeline uses pictures as symbols along the timeline as illustrations of the event being depicted. To create this graphic, the normal plot axes, values and symbols are suppressed. The images are matched to the data points in an annotate dataset using the image value of the annotate function variable. The bottom axis label represents the formatted value of the timeperiod variable. When using images, the most important change from using symbols or fonts is that it is necessary to create space for the image. In this example, the space is created by expanding the axes and by adding to the x and y variables after reading in the images.



EXAMPLE 1b SELECTED SAS CODE

```
/*Create annotate data set to set images as plot points*/
data images;
  length function $ 8 style $ 15 imgpath $200.;
  retain xsys ysys '2' hsys '3' when 'a' position '4' style 'fit';
set health;
x=timeperiod-.5; y=level-1;
if timeperiod=1 then do;
  function='move'; output;
  imgpath='r:\bokerson\sesug 2011\med1.bmp'; x=x+1.1; y=y+1;
  function='image'; output; end;
else if timeperiod=2 then do;
  function='move'; output;
  imgpath='r:\bokerson\sesug 2011\med2.bmp'; x=x+1.1; y=y+1;
  function='image'; output; end;
else if timeperiod=3 then do;
  function='move'; output;
  imgpath='r:\bokerson\sesug 2011\med3.bmp'; x=x+1.1; y=y+1;
  function='image'; output; end;
else if timeperiod=4 then do;
  function='move'; output;
  imgpath='r:\bokerson\sesug 2011\med4.bmp'; x=x+1.1; y=y+1;
  function='image'; output; end;
run;

/*Set Symbols and Axes*/
symbol value=none;
axis1 split='/' label=none minor=none major=none style=0 order=(0 to 5 by 1);
axis2 label=none minor=none major=none order=(0 to 4 by 1)
value=none style=0;
```

```

/*Plot the timeline*/
proc gplot data=health;
plot level * timeperiod /annotate=images
      haxis=axis1 vaxis=axis2 noframe;
format timeperiod hlthevent.;
run;
quit;

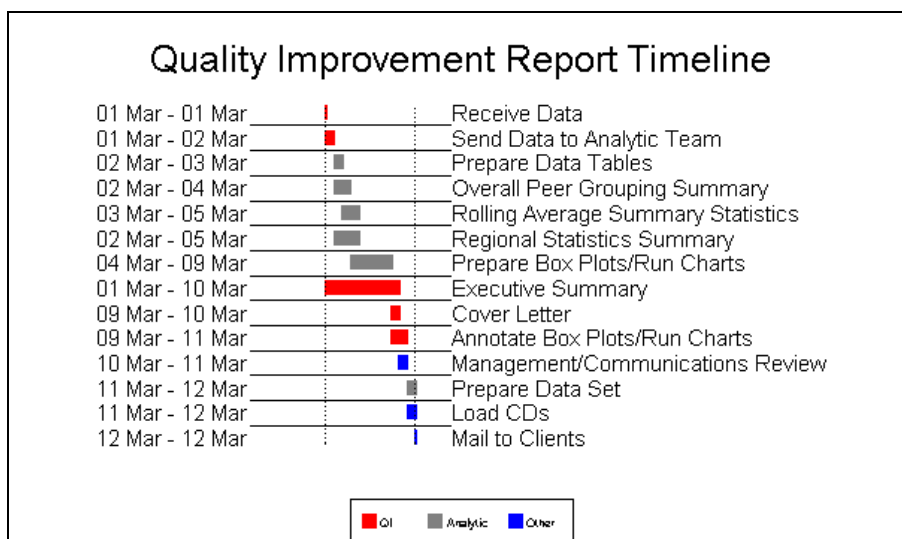
```

EXAMPLE TWO: PROJECT TIMELINES

Project management requires the planning, organizing, securing and managing of people and resources that are necessary for successful completion of specific project goals and objectives within the desire timeframe. Project timelines are used to identify and manage all the time and resources needed for project completion. The two most important uses for project timelines are to: 1) explain your project to strangers, and 2) organize your project for you. SAS software system provides all the tools for project timelines that meet these two needs, by allowing the display on a single page. SAS supports time scheduling and display, through both traditional and specialty products. While many external specialty software products also exist for project management. SAS provides a solution that is robust enough that the actual format of the timeline is up to the user. Two examples follow.

EXAMPLE 2a– TASK TIME SCHEDULE

In this example, a time schedule is created for a task within a project. Areas of responsibility are color-coded. This type of schedule is particularly useful because it provides, at a glance, the same information as well-known project planning software. The display is limited to the most necessary factors and provides the time schedule in an easy-to-read, scalable format.



EXAMPLE 2a SELECTED SAS CODE

```

/*Create Annotate Data Set to Draw Lines*/
data anno;
  length color $8 ;
  retain xsys ysys '2' size 10;
  set process;
  line=linetype; color=color;
function='move'; x=end+.3; y=nobs; output;
function='draw'; x=begin; y=nobs; output;
run;

/* Set Titles Symbols and Axes*/
title1 h=4 c=black "Quality Improvement Report Timeline";
symbol1 i=none;
axis1 label=none major=none minor=none style=0 ;
axis2 label=none major=none minor=none style=0 order="20Feb10"d to "16Mar10"d

```

```

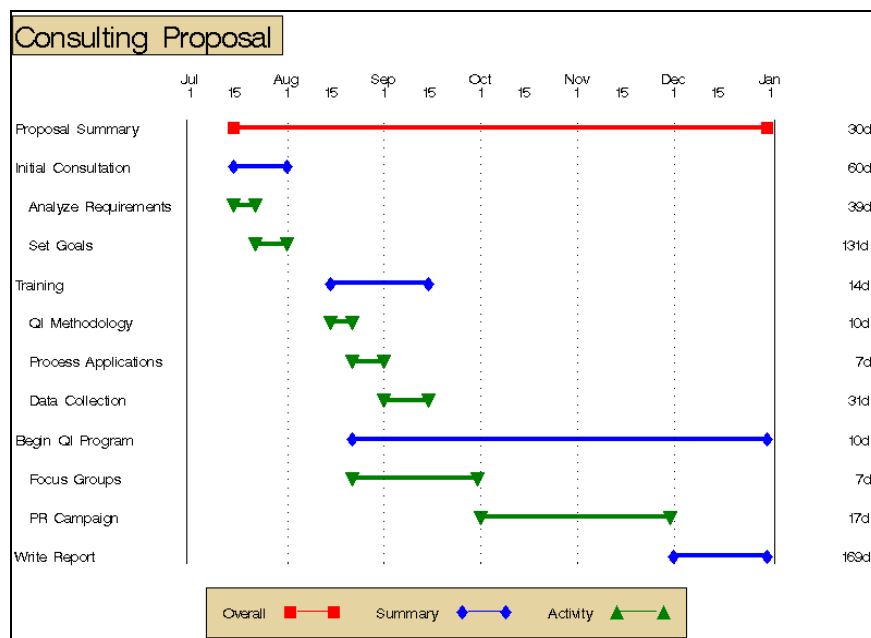
value=none length=45;
axis3 label = none value=(j=1 %longtasks) major=none minor=none style=0 ;

/*Plot the Timeline*/
proc gplot annotate=anno;
plot nobbs*end/
    vaxis=axis1 haxis=axis2 lhref=33 href="01MAR10"d "12MAR10"d vreverse;
plot2 nobbs*end/vaxis=axis3 haxis=axis2 vreverse;
format nobbs order.;
footnote2 box=1 f=marker c=red h=1.5 'U' f='Arial' c=black h=1.5 ' QI '
    f=marker c=gray h=1.5 'U' f='Arial' c=black h=1.5 ' Analytic ' f=marker
    c=blue h=1.5 'U' f='Arial' c=black h=1.5 ' Other ' ;
run;
quit;

```

EXAMPLE 2b – PROJECT PROPOSAL TIMELINE

This example provides a one-page activity proposal for a consulting contract. Categories and activities within categories are displayed on a timeline, with the number of days for each activity also displayed. Like the previous timeline, the code, once developed, can serve as a template for future proposals. Although the graphic as displayed shows all categories and subcategories, the code allows for alternate display of major categories only. Additional color-coding can be added to identify areas of responsibility. Proposal section numbers can also be added as left axis values if required.



EXAMPLE 2b SELECTED SAS CODE

```

/*Create task order format*/
data addtaskn;
set process; taskord=_n_; format task $char22.; run;
data taskordr;
set addtaskn(rename=(taskord=starttask=label)); fmtname='dataord'; type='N';
keep fmtname label start type;
run;
proc format cntlin=taskordr;run;

/*Create Annotate Data Set to Draw Lines*/
data anno;
length function color $8; retain xsys ysys '2';
set addtaskn; line=linetype;
function='move'; x=begin; y=taskord; output; function='symbol';

```

```

    style="marker";    size=1;
if taskord=1 then do; text='U'; color="red"; end;
    else if substr(task,1,1)=" " then do;text="D"; color="green"; end;
    else do; text='P'; color="blue"; end; output;
function='draw'; size=4; x=end; y=taskord; output;
function='symbol'; style="marker"; size=1; end;
if taskord=1 then do; text='U'; color="red"; end; output;
run;

/*Label top axis*/
data anno2; set dates(keep=date);
    length function text $8 ;
    retain xsys '2' ysys '3' color 'black' when 'a' size 1; line=1;
    monname=put(date,monname3.);
function='move'; x=date; y=90; output;
function='label'; x=date; y=90; text=left(monname); output;
function='label'; x=date; y=88; text=left(day(date)); output;
run;

/* Set Titles, Symbols and Axes*/
title j=1 box=1 bspace=0 bc=paoy
"Consulting Proposal"; title3 " ";
axis1 label=none major=none minor=none value=(j=1 h=1);
axis2 minor=none label=none major=none value=none style=0
    order=("01Jul10"d to "01Jan10"d by month);
axis3 major=none minor=none label=none value=(j=1 %days);
symbol1 value=none;

/*Plot the Timeline*/
proc gplot annotate=anno data=addtaskn;
plot taskord*end/frame vreverse vaxis=axis1 haxis=axis2 caxis=black
    anno=anno2 href="01Aug10"d "01Sep10"d "01Oct10"d "01Nov10"d "01Dec10"d
    lhref=34 cframe=white;
plot2 taskord*end/vaxis=axis3 vreverse;
format taskord dataord. end mmddyy10.;
footnotel c=black 'Overall ' c=red font=marker height=1 'U' move=(-1,+0)
    draw=(+0,+0.5,+1cm,+0) move=(+3.5,+0) "U" f=swiss c=black 'Summary ' c=blue
    font=marker height=1 'P' move=(-1,+0) draw=(+0,+0.5,+1,+0) move=(+3.5,+0) "P"
    f=swiss c=black 'Activity ' c=green font=marker height=1 'D' move=(-1,+0)
    draw=(-1,+0.5,+1cm,+0) move=(+3.8,+0) "D" c=black bc=paoy box=1 bspace=1;
run;
quit;

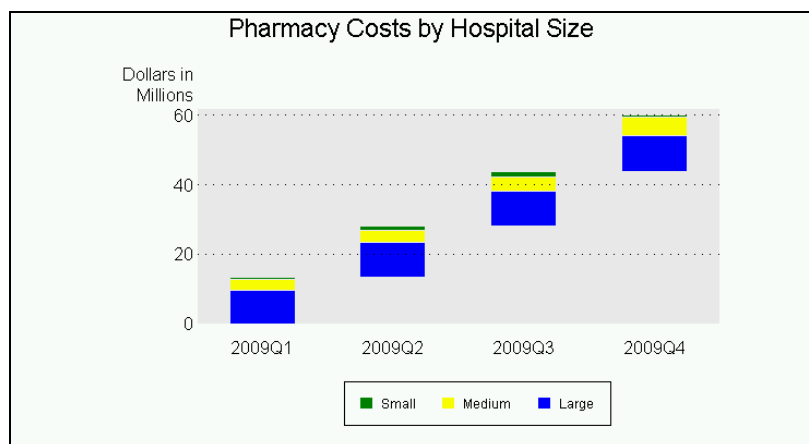
```

EXAMPLE 3: WATERFALL CHARTS

The waterfall chart is ideal for displaying areas of loss or gain between starting and ending values. In general use, waterfall charts show how an initial value is increased and decreased by a series of intermediate values, leading to a final value. The advantage of displaying this type of information in a waterfall chart over a stacked bar chart is that, by isolating the subcategories of interest, the viewer's eye is drawn to the highlighted information. The waterfall chart is a relatively new graphic that has recently been popularized by McKinsey and Company for illustration of quantitative flows. In this paper, waterfall charts are created with two different methodologies.

EXAMPLE 3a: PHARMACY EXPENDITURE WATERFALL

In the first example, the waterfall chart shows the contributions to pharmacy costs by small, medium and large hospitals in a hospital system. This type of graph is often accomplished by creating an invisible column that keeps the increases and decreases linked to the heights of the previous columns. This method is the standard for Excel and is also demonstrated in a recent SESUG paper using Proc GHART to show cumulative campaign contributions (Booth, 2008).



EXAMPLE 3a SELECTED SAS CODE

```

/*Set Patterns*/
pattern1 v=s c=cxe8e8e8;
pattern2 v=s c=blue;
pattern3 v=s c=yellow;
pattern4 v=s c=green;

/*Define axis characteristics */
axis1 major=none minor=none order=0 to 60 by 20 c=white value=(c=black)
label=(c=black "Dollars in" j=r "Millions");

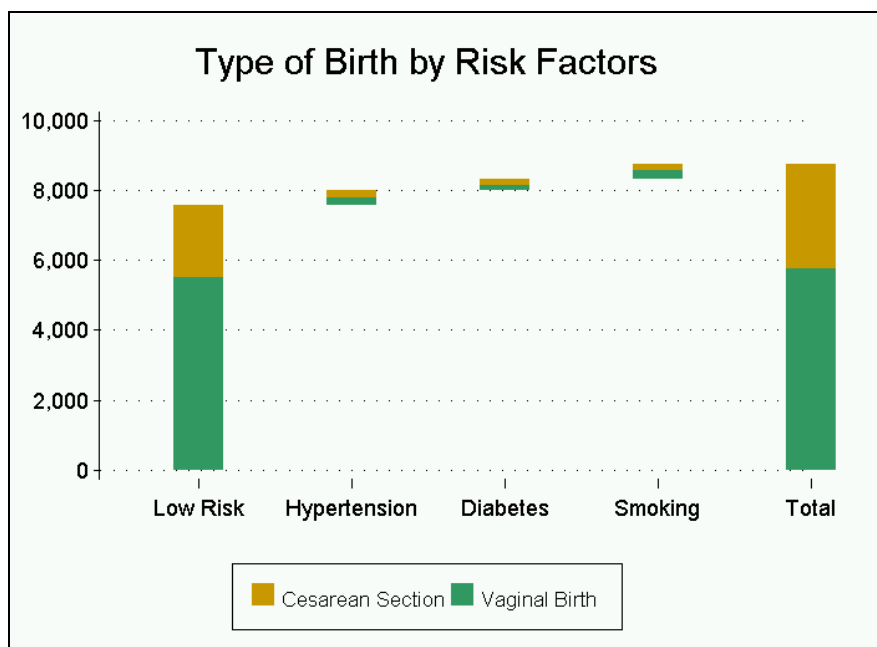
/*Add the title and title options*/
goptions ftitle='Arial' ftext='Arial' htitle=2 htext=1.4;run;
title 'Pharmacy Costs by Hospital Size';

/*Produce the Waterfall Chart */
proc gchart data=hospitals;
vbar period / discrete sumvar=cost
    raxis=axis1 autoref lref=34 cref=black subgroup=class
    cframe=cxe8e8e8 width=15 space=15 coutline=cxe8e8e8 nolegend;
format period yyq6. cost dollar4.0;
footnotel move=(-12,0) f='Arial' h=1.1 box=1 c=green f=Marker 'U' c=black
f='Arial' ' Small' c=yellow f=Marker ' U' c=black f='Arial' ' Medium'
c=blue f=Marker ' U' c=black f='Arial' ' Large';
run;
quit;

```

EXAMPLE 3b: HIGH RISK BIRTH WATERFALL

In this example, the waterfall chart shows a standard risk population, with the intermediate values as high risk populations. Waterfall charts do not need to have invisible columns. Here the bars are created in the annotate data set, which allows the beginning and ending points to be data driven. Because the bar is wider than a data point, it is necessary to use the offset option in the horizontal axis statement to move the starting bar to the right; otherwise the bar will overlap the vertical axis line. This example illustrates a waterfall chart with two categories. To add additional categories and colors, just add these to the color statement in the annotate data step. Regardless of the number of categories, no invisible column is created. For this reason, the GPLOT procedure is the base rather than GCHART. This waterfall chart also differs from the earlier one in that the final bar is a summary bar, a common inclusion in waterfall charts.



EXAMPLE 3b SELECTED SAS CODE

```
/*Create the annotate data set*/
data anno;
  length function color $8;
  retain xsys ysys '2' size 40 color 'cx66CC33' when 'a';
  set waterfall;
  line=line;
  if type="C" then color='cxFF9900';
  else color='cx66CC33';
  function='move'; y=begin; xc=Risk; output;
  function='draw'; y=end; xc=Risk; output;
run;

/*Set Titles, Symbols and Axes*/
title "Type of Birth by Risk Factors";
symbol1 i=none;
axis1 order=("Low Risk" "Hypertension" "Diabetes" "Smoking" "Total")
  offset=(10,0) label = none style=0 ;
axis2 Order=(0 to 10000 by 2000) minor=none label=none;

/*Plot the Waterfall*/
proc gplot annotate=anno;
  plot end*Risk/haxis=axis1 vaxis=axis2 autovref lvref=34 hminor=0;
  format end comma6.;
run;
quit;
```

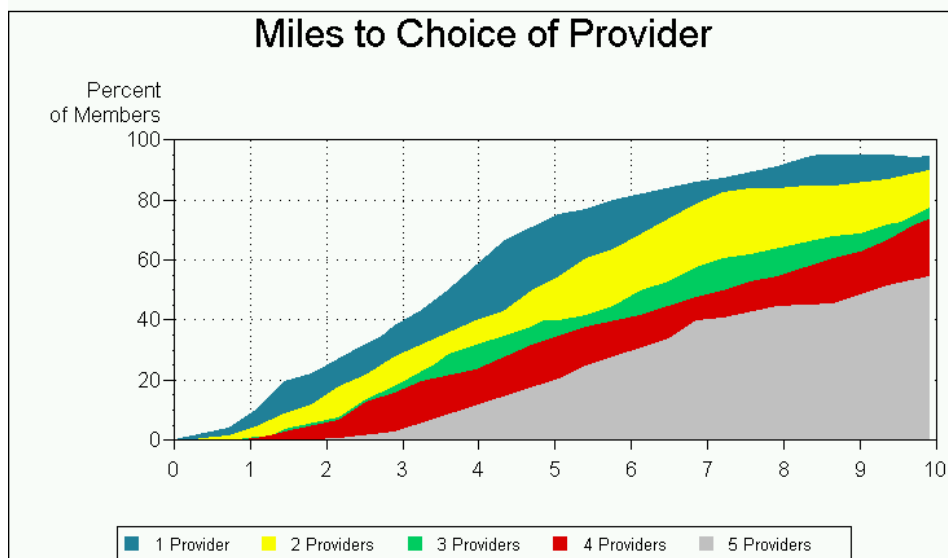
EXAMPLE FOUR – DISTANCE CHARTS

There is great advantage to tying health care data to geographic coordinates, but all distance mappings in the health care arena do not need radians or latitude/longitude to provide meaningful information. The examples below are illustrations of non-geographic distance mapping.

EXAMPLE 4a: AREA DISTANCE PLOT

In this example, the area plot option of SAS Proc Gplot is used to create a distance map without reference to a specific geographic place. Geocoding (Proc GEOCODE) was used to record coordinates of plan member's residences so that they could be used to calculate the distance from network providers. The code used in this example uses the plus4 option. The lookup table for this option is not shipped automatically by SAS and needs to be

downloaded or obtained from another source. As in several of the other examples presented here, the legend is generated in a footnote statement as legends are not yet supported with multiple plot statements.



EXAMPLE 4a SELECTED SAS CODE

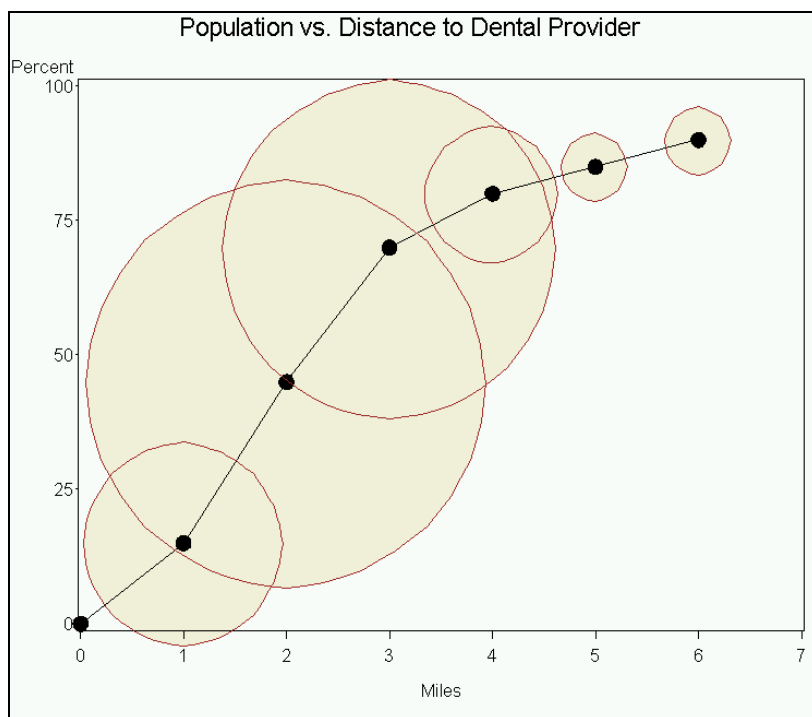
```
/*Geocode for distance calculation*/
proc geocode
plus4
lookup=lookup.zip4
data=work.members out=work.geo_members
run; quit;

/*Plot the distance map*/
proc gplot data=provider2;
plot p5*interval p4*interval p3*interval p2*interval p1*interval /overlay
autovref autovhref cvref=black chref=black lautovref=34 lautohref=34
haxis=axis1 vaxis=axis2 caxis=black vminor=3
areas=5;

footnote2 box=1 f=marker c=ltgray 'U' f='Arial' c=black ' 1 Provider '
f=marker c=cxFFFF00 'U' f='Arial' c=black ' 2 Providers '
f=marker c=cx00CC66 'U' f='Arial' c=black ' 3 Providers '
f=marker c=cxD80000 'U' f='Arial' c=black ' 4 Providers '
f=marker c=cx7Ba7E1 'U' f='Arial' c=black ' 5 Providers ';
run;
quit;
```

EXAMPLE 4b – DISTANCE BUBBLE PLOT

It is often important to understand what proportion of a population is reached within a certain distance or mileage. In this additional example of provider to plan member distance, the bubbles represent the proportion of the population at that distance, while the points represent the cumulative percentage at that distance. While bubble plots are a bit different than the other graphics in this paper, as SAS does directly create a bubble plot, using the bubble plot in combination with geocoded data as a distance chart is a custom use of this graphic.



EXAMPLE 4b SELECTED SAS CODE

The code used to create this graphic uses two annotate data sets, one for the circle and the other for the circle border, since the circle is plotted before the plot data set (dental) and the border after the plot data set.

```
data annoplot; set dental;
  length function $8 text \ $1;
  retain XSYS YSYS '2' style 'Marker' function 'LABEL'
    when 'b' color 'cxF2F2DF';
  size = size;
  X=distance; y=population;
  text='W';
output;
run;

data annoplot2; set dental;
  length function $8 text $1;
  retain XSYS YSYS '2' style 'Markere' function 'LABEL'
    when 'a' color 'brown';
  size = size;
  X=distance; y=population;
  text='W';
output;
run;
goptions ftext='Arial' ftitle='Arial' htext=1.4;
title h=2 'Population vs. Distance to Dental Provider';
axis1 order=0 to 7 by 1 minor=none label=('Miles');
axis2 minor=none order= 0 to 100 by 25 label=('Percent');
symbol i=join v=dot h=2;

proc gplot data=dental anno=annoplot;
  plot population*distance/
    vaxis=axis2 haxis=axis1 anno=annoplot2;
run;
quit;
```

CONCLUSION

This paper illustrates the built-in flexibility of SAS that allows the creation of custom graphics and charts. These graphics go a step beyond the customization of existing charts, yet can be created using the standard SAS graphics procedures. By using the built-in flexibility of SAS/GRAPH, as well as the capabilities of the SAS annotate facility and SAS/GRAPH procedures, SAS can be used to create whatever graphical display is optimum for displaying data and what that data means. Although not covered in this paper, additional flexibility is available through SAS/GRAPH SG procedures, ODS statistical graphics, ODS graphic editor, ODS graphics designer, Enterprise Guide and other SAS products.

REFERENCES

- Booth, RB. "Ride the Wave – Using Waterfall Graphs to Easily Depict Gaps between Starting and Ending Values." *SESUG Proceedings*. 2008.
- Hadden, Louise. "Behind the Scenes with SAS®: Using Custom Graphics in SAS Output." SAS Global Forum 2009.
- Massengill D, Odom E. "PROC GEOCODE: Creating Map Locations from Your Data." SAS Global Forum 2009.
- Okerson, Barbara B. "Distance Mapping in Health and Health Care: SAS® as a Tool for Health Geomatics", *SESUG Proceedings*, 2010.
- Okerson, Barbara B. "Scheduling Time with SAS®: Project Proposal Examples" SUGI 28 Proceedings. SAS Institute Inc. Cary, NC. 2002.
- Okerson, Barbara B. "Using SAS to Create Custom Graphics." SAS Global Forum, 2011.
- Raisal, EM. *The McKinsey Way: Using the Techniques of the World's Top Strategic Consultants to Help You and Your Business*. McGraw-Hill Professional, 1999.
- Robbins, Naomi B. 2005. *Creating More Effective Graphics*. John Wiley & Sons, Inc., New Jersey.
- SAS Institute Inc. "New Features and Enhancements in SAS 9.2 SAS/GRAPH® Software." http://support.sas.com/kb/31/addl/fusion_31477_1_enhancementsandnewfeaturesinsas92robertallison.pdf Accessed February 2011.
- SAS Institute Inc., SAS Online Doc v.9, <http://support.sas.com/onlinedoc/913/docMainpage.jsp>, 2007.
- SAS Institute Inc. Technical Papers and Presentations. <http://support.sas.com/rnd/papers>. Accessed October 2010.
- Zdeb, Mike and Allison, Robert. "Stretching the Bounds of SAS/GRAPH Software." SUGI 30. April 2005.

ACKNOWLEDGEMENTS

A special thank you goes to Marcia Surratt from SAS Technical Support for some timely tips on incorporating images into graphics.

Historical medical images are used by permission of U.S. National Library of Medicine and can be found at: www.nlm.nih.gov/hmd/ihm/.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. For more information contact:

Barbara B. Okerson, Ph.D., CPHQ, FAHM
 SESUG 2011 Academic Conference Chair
 Senior Health Information Consultant,
 West Region Client Advisory & Reporting Services
 WellPoint Enterprise Information Management
 8831 Park Central Drive, Suite 100, Richmond, VA 23227
 Office: 804-662-5287 Fax: 804-662-5364
 Email: barbara.okerson@wellpoint.com

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.