Paper 135-28

What's In A Map?

A Macro-Driven Drill-Down Geo-Graphical Representation System

Louise Hadden, Abt Associates Inc., Cambridge, MA

ABSTRACT

This paper presents a macro-driven system to create drill-down maps and present them on intranets or the internet, using BASE SAS® and SAS/GRAPH®. This system can be used with SAS V8 (or SAS V612 in conjunction with the %ds2htm macro supplied by the SAS Institute for HTML only) in any operating environment, including Red Hat Linux, Windows, MVS/OS390, and AIX/UNIX. Although the primary focus of the paper is on interactive HTML map output, some different graphic outputs (for PC SAS only) will be briefly demonstrated. Presentation graphics contained in HTML, XML and PDF files are static and require only an internet browser and/or Adobe Acrobat to view. They are quickly and easily updatable on the host system even if the server does not have SAS installed. Other presentation options SAS provides such as JAVA and Active-X are interactive, but require specific versions of SAS or special software add-ons supplied with specific versions of SAS to be installed on the host system to allow interactivity.

INTRODUCTION

Since I first started using a beta version of SAS on an IBM 360 at MIT to analyze data for a thesis more than 20 years ago, SAS has evolved into an amazing computing tool. The purpose of this paper is to outline a simple method to take advantage of the new and improved graphic functionality in SAS V8 using some old tried and true tools such as SAS Macro Language and user-defined formats, and some new methods provided by the Output Delivery System (ODS). Some of the presentation options demonstrated in the paper are specific to PC-SAS; others can be used and have been tested on other platforms including AIX UNIX and Red Hat Linux.

KNOW YOUR DATA (GETTING UP CLOSE AND PERSONAL WITH SAS-SUPPLIED MAP DATA SETS)

The map-related data sets provided by SAS have always seemed somewhat mysterious to me, possibly because as an "old dog" I have not been accustomed to thinking of computing in a graphic way. From paper tape and punch cards to GUI is a long way to travel! My approach to understanding the map data sets is the same approach I use with any "unknown" data set; do a proc contents and test print some observations and take a look.

SAS provides maps and map-related data bases for geographical areas all over the world within the SAS/GRAPH package. In the US these maps are free of charge down to the STATE COUNTY level. Should you wish to represent finer areas using SAS, you have the choice of purchasing specific map sets from SAS or other vendors, or creating your own SAS map data sets yourself (I'll leave a description of this process for a future paper!)

Map data sets provided by SAS can be "projected" or "unprojected". Projected map data sets have had their longitude and latitude converted to X and Y coordinates and have been "flattened" so as to present well on paper. While you can (and sometimes have to) re-project map data sets, the BEST representations come from projected map data sets that SAS provides. (For example, SAS has figured out the best angle of projection to represent the entire United States, and added Alaska and Hawaii to the map.) If you want to subset a particular state from the U.S. map, you must re-project the map to create and/or adjust the X and Y coordinates. Below follows the output of a proc contents and test print for the SAS US map (maps.us.)

```
SAS-Supplied Map Data Sets
US Map Contents

The CONTENTS Procedure

Data Set Name: MAPS.US                 Observations:     1525
Member Type:   DATA                    Variables:           4
Engine:        V8                      Indexes:             0
Created:       8:50 Thursday,          Observation Length: 22
               June 8, 2000
Last Modified: 8:50 Thursday,          Deleted Observations:0
               June 8, 2000
Protection:                            Compressed:         NO
Data Set Type:                         Sorted:             NO
Label:         United States,reduced-projected:
               Copyright(C) 2000 SAS Institute Inc.

-----Engine/Host Dependent Information-----

Data Set Page Size:          4096
Number of Data Set Pages:    9
First Data Page:             1
Max Obs per Page:            184
Obs in First Data Page:      115
Number of Data Set Repairs:  0
File Name: C:\Program Files\SAS
Institute\SAS\V8\maps\us.sas7bdat
Release Created:             8.0202M0
Host Created:                WIN_NT

-----Alphabetic List of Variables and Attributes-----

#    Variable    Type    Len    Pos    Label
------------------------------------------------------------
2    SEGMENT     Num       5      5    State Segment Number
1    STATE       Num       5      0    State FIPS Code
3    X           Num       6     10    X Coordinate
4    Y           Num       6     16    Y Coordinate

US Map Test Print

STATE    SEGMENT       X           Y

  1          1       0.16175    -0.10044
  1          1       0.12305    -0.10415
  1          1       0.12296    -0.10678
  1          1       0.12667    -0.11010
  1          1       0.12629    -0.11467
```

As you can see, this projected map data set contains many points represented by X and Y coordinates, associated with states and segments within states.

Some "map" data sets SAS provides contain names of areas, associated with geographical points designated by both X and Y coordinates matching "projected" map data sets, and longitude and latitude matching "unprojected" map data sets. Below follows a proc contents and test print from a map data set that provides center points for US states.

```
US Center Map Contents

The CONTENTS Procedure

Data Set Name: MAPS.USCENTER             Observations:     60
Member Type:   DATA                      Variables:         6
Engine:        V8                        Indexes:           0
Created:       15:04 Tuesday,          Observation Length:  30
               August 10, 1999
Last Modified: 15:04 Tuesday,          Deleted Observations: 0
               August 10, 1999
Protection:                              Compressed:       NO
Data Set Type:                           Sorted:           NO
Label:         Copyright(C) 1988 SAS Institute Inc. USA


-----Engine/Host Dependent Information-----

Data Set Page Size:          4096
Number of Data Set Pages:    1
First Data Page:             1
Max Obs per Page:            135
Obs in First Data Page:      60
Number of Data Set Repairs: 0
File Name: C:\Program Files\SAS
Institute\SAS\V8\maps\uscenter.sas7bdat
Release Created:             8.0000M0
Host Created:                WIN_NT


-----Alphabetic List of Variables and Attributes-----

#  Variable  Type  Len  Pos  Label
------------------------------------------------------------
4  LAT       Num     6   12  Unprojected Latitude in Degrees
3  LONG      Num     6    6  Unprojected Longitude in Degrees
1  OCEAN     Char    1    0  Y or N
2  STATE     Num     5    1  State FIPS Code
5  X         Num     6   18  X Coordinate
6  Y         Num     6   24  Y Coordinate

SAS-Supplied Map Data Sets
US Center Map Test Print

OCEAN     STATE      LONG       LAT          X           Y

  N         1       86.600    32.8000     0.13511    -0.07158
  N         2      152.000    65.0000    -0.31866    -0.12302
  N         4      111.500    34.5000    -0.22318    -0.02998
  N         5       92.500    35.0000     0.04761    -0.03877
  N         6      120.000    37.0000    -0.33122     0.03774
```

A vital step in using SAS/GRAPH to create maps is linking your data to the particular map data set you wish to use. It is even more important to be knowledgeable about the linking variable(s) when producing "interactive" HTML. The linking variable we will be using to produce our interactive map of the United States is STATE, a numeric variable with length of 5 which is the State FIPS Code. Thus, any data set(s) we wish to represent on the map must have a numeric variable called STATE with a length of 5.

BUILDING A CUSTOM MAP

Although the SAS-supplied maps are adequate for most purposes as they stand, you may want to build a custom map. For example, you might want to present an analysis of variables based on counties in the state of Montana only, or you might want to label states on the US map. Below follow two examples: one is to create a county map of Montana, and the other is to "annotate" the US map or label states.

Note that all the examples following do not represent entire programs, but merely portions of programs. Please e-Mail the author for complete programs. Note also that proper indentation has been altered in order to fit into the paper format.

```
?* Program to create a map of Montana */

goptions reset=global;

/* select Montana records from US County map */

proc sort data=maps.county (where=(state eq 30)) out=mt1;
    by state county;
run;

/* reproject the map of Montana */

proc gproject data=mt1 out=mt2;
    id state county;
run;

/* . . . more code . . . */
/* use the map */

proc gmap data=county map=county2;
    id state county;
    choro countynm / discrete  html=linkme
        coutline=gray nolegend annotate=anno2;
title1 f=swissb h=1 "Montana County Data";
run;

quit;

/* Program to create an annotate data set for the map of the
United States created in PUTTING IT ALL TOGETHER below */

goptions reset=global;

/* annotate data set for state names */

data mapanno (drop=ocean long lat state);
      length text $ 20 color function style $ 8;
      retain function 'label' color 'black' xsys ysys '2' size 2
             hsys '3' style 'zapfb' size 2 position '5' when 'a';
        set maps.uscenter (where=(state le 56 and ocean ne 'Y'));

        text=fipstate(state);
run;
```

If you want to annotate a map of areas that do not have pre-calculated centers as in the data set USCENTER accessed above, SAS has provided two annotate macros to perform centroid calculations, %CENTROID and %MAPLABEL. %CENTROID calculates the centroid of a polygon (if you've ever tried to do this yourself, you'll understand what a blessing this is!) and %MAPLABEL defines a label on the centroid. %MAPLABEL calls the %CENTROID macro, so both need to be submitted for this process to work. These macros can be made available in Version 9 by first submitting %ANNOMAC within your SAS program. If you are using a prior version of SAS, the macros can be found at http://ftp.sas.com/techsup/download/graph/maplabelmacros.txt and the code for both macros needs to be submitted prior to use in a program.

BUILD YOUR PRESENTATION DATA SET(S)

Note: for the purposes of this paper all analytic data elements in the example shown have been scrambled to protect data confidentiality.

The most important aspect to building a presentation data set to be used with a map is to correctly "link" items to the geographic areas being used. In the next example shown, I have used a format to associate values for the variables STATE and QINAME with an HTML reference, which can then be used in a data step or procedure later. This could easily be done in a data step with a permanent format being stored following the use of proc format cntlin, but for the purposes of demonstrating the linking process it is shown the "long" way below.

```
/* create formats for html drilldown */

proc format;
/* link states on map */
    value ststat
          1='href="1.htm"'
          2='href="2.htm"'
          and so on . . .

         56='href="56.htm"'
         ;
/* link listings to header under state */
    value $qinamef
    "COCADL1"="<A href=COCADL1.htm>CADL1</A>"
    "COCDRG1"="<A href=COCDRG1.htm>CDRG1</A>"
     and so on . .
    "WAPWALX"="<A href=WAPWALX.htm>PWALX</A>"
    other="NO DATA"
    ;
run;
```

The next step is to write a macro to create data tables and/or graphics which can be associated with the formatted HTML references created above.

```
/* this creates the reports for each state and qiname */

ods listing close;

%macro runit(stcode,qiname,qilab,stname);

ods html body="&stcode.&qiname..htm" style=styles.sasweb;

data &stcode.&qiname;
    length qiname $ 5 qilabel $ 40;
    set dd.q4_comb (keep=st_code fac_itl name fac_city
                         mcare_id aj_&qiname os_&qiname
                         ds_&qiname ns_&qiname
                         ajs&qiname ajn&qiname oss&qiname
                         osn&qiname
                  rename=(aj_&qiname=ADJUSTED
                          ds_&qiname=DENOMNTR
                          ns_&qiname=NUMERATR
                          os_&qiname=OBSERVED
                          oss&qiname=obsstmn
                          osn&qiname=obsallmn
                          ajs&qiname=stfacmn
                          ajn&qiname=allfacmn)
                  where=(st_code="&stcode."));
    qiname="&qiname";
    qilabel="&qilab";
/* relabel and label variables */
label qiname='Quality Indicator Acronym'
          and so on . . .
      allfacmn='All State Adjusted Mean';
    format adjusted stfacmn allfacmn observed obsstmn
          obsallmn 7.1 denomntr numeratr comma7.;
    format fac_itl $char10. mcare_id $char6.;

```

```
/* here's a bunch of call symputs to do the titles */

call symput('adjstmn',left(put(stfacmn,7.1)));
call symput('adjusmn',left(put(allfacmn,7.1)));
call symput('obsstmn',left(put(obsstmn,7.1)));
call symput('obsusmn',left(put(obsallmn,7.1)));

run;

proc print data=&stcode.&qiname label uniform;
   var name fac_city mcare_id denomntr numeratr observed
      adjusted;
   id fac_itl;
title1 "&stname / &qilab  QI NAME=&qiname - All Data
Randomized";
title2 "STATE MEAN (OBSERVED) = &obsstmn";
title3 "SIX-STATE MEAN (OBSERVED) = &obsusmn";
title4 "STATE MEAN (ADJUSTED) = &adjstmn";
title5 "SIX-STATE MEAN (ADJUSTED) = &adjusmn";
footnote1 "ALL DATA RANDOMIZED TO PROTECT CONFIDENTIALITY";
run;

ods html close;

%mend;

%RUNIT(CO,CADL1,CC: LATE-LOSS ADL WORSENING,COLORADO);
%RUNIT(CO,CDRG1,CC: PREV OF ANTIPSYCH DRUG USE,COLORADO);
%RUNIT(CO,CINFX,CC: INFECTIONS PREVALENCE,COLORADO);
and so on . . .
%RUNIT(WA,PWALX,PAC: IMPROVEMENT IN WALKING,WASHINGTON);
```

Example:

WASHINGTON / CC: LATE-LOSS ADL WORSENING QI NAM
- All Data Randomized
STATE MEAN (OBSERVED) = 6.5
SIX-STATE MEAN (OBSERVED) = 6.6
STATE MEAN (ADJUSTED) = 6.3
SIX-STATE MEAN (ADJUSTED) = 6.4

MDS System Internal Facility Identifier	Facility Operating Name	City Where Facility Is Located	Medicare (OSCAR) Provider Number	# of Residents after exclusions/missing covariates
0000000093	ALDERCREST HEALTH & REHAB CENT	EDMONDS	505236	77
0000000094	ALDERWOOD MANOR	SPOKANE	505257	57

Note: Graphic is cropped to save room in the paper.

```
/* this creates the graphic link (itself an HTML table) from each
state to the reports for each state and quality indicator */

    data dd.stlink;
        length linkme $ 80;
        set ee.stqi;
        linkme=put(state,ststat.);
        label QI_LABEL='Quality Indicator Description'
              STQI='Quality Indicator Acronym'
              QI_NAME='Quality Indicator Acronym';
run;

    %macro details(st,sttit);

    ods html body="&st..htm" path=odsout;
    title1 "Quality Indicators for &sttit";
    proc print data=dd.stlink (where=(state=&st)) noobs label
    uniform;
        var STQI QI_LABEL;
    format STQI $qinamef.;
    run;

    ods html close;

    run;

    %mend details;

    %details(1,Alabama);
    %details(2,Alaska);
    and so on . . .
    %details(56,Wyoming);
```

Example:

Quality Indicators for Washington

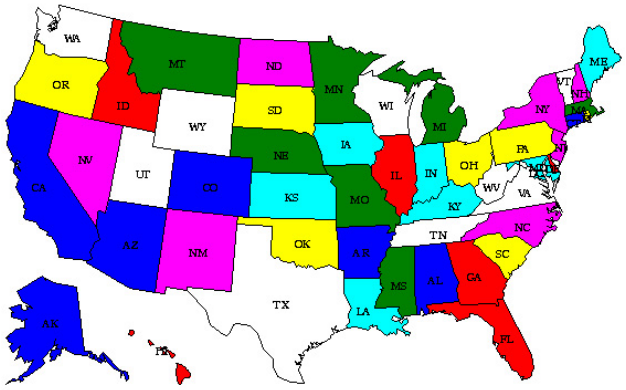
Quality Indicator Acronym	Quality Indicator Description
CADL1	CC: LATE-LOSS ADL WORSENING
CDRG1	CC: PREV OF ANTIPSYCH DRUG USE
CINFX	CC: INFECTIONS PREVALENCE
CPAIX	CC: PAIN--INADEQUATE MANAGEMENT
CPRU1	CC: PREV OF PRESSURE ULCERS
CRES1	CC: PREV OF RESTRAINTS USED DAILY
CWGT1	CC: WEIGHT LOSS PREVALENCE
PDELX	PAC: FAILURE TO IMPROVE/MANAGE DELIRIUM
PPAIX	PAC: INADEQUATE PAIN MANAGEMENT
PWALX	PAC: IMPROVEMENT IN WALKING

PUTTING IT ALL TOGETHER

The portion of code presented below creates a "clickable" HTML document or map of the United States. Note the use of the annotate data set created above to label the states with their postal abbreviations.

```
goptions device=gif ftext=zapfb htitle=3 gunit=pct hsize=6 in
vsize=5 in;

    pattern1 value=msolid color=blue repeat=7;
    pattern2 value=msolid color=red repeat=7;
    pattern3 value=msolid color=cyan repeat=7;
    pattern4 value=msolid color=green repeat=7;
    pattern5 value=msolid color=magenta repeat=7;
    pattern6 value=msolid color=yellow repeat=7;
    pattern7 value=mempty repeat=56;

    ods html body='maphead.htm' path=odsout;

    proc gmap data=dd.stlink map=maps.us;
        id state;
        choro state / discrete html=linkme coutline=black
                      nolegend annotate=mapanno;
    title1 'Quality Indicators by State';
    run;
    quit;
```

Quality Indicators by State

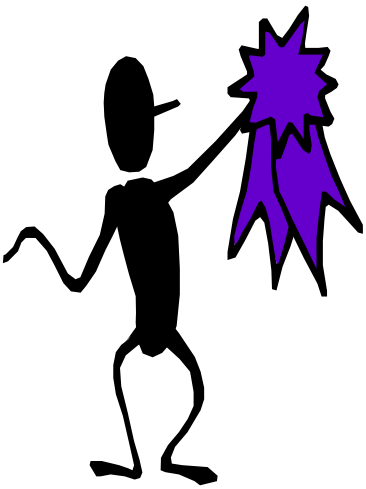


The idea is that one can click on any state on the map to get more information about that state. In this case, I have scrambled the confidential data that was actually produced for a contract so that the reports presented are not "real". If you open up the HTML map in a browser and click on the state of Washington in the map, you see the HTML table titled "Quality Indicators for Washington". If you then click on one of the items in the Quality Indicator Acronym in the HTML table, you then get a table similar to the [cropped] table on ADL worsening for Washington above. The number, content and format of HTML "layers" is entirely up to you, the programmer. Please note, however, that each "layer" may contain many html files; space and naming conventions may become an issue!

The resolution of the graphics shown above suffer when they are compressed into a size that fits into a paper format and printed in black and white, but the graphics and reports are quite striking in their full, interactive glory on the computer screen.

I will have sample maps and programs available on a laptop at my poster during "Meet the Presenters" session. Please note that the program portions presented above are incomplete and cannot be run "as is".

TOOL TIPS



In the portion of the program used in "Putting It All Together" you will notice the html option in the choro statement, referring to a variable which contains a hard-coded HTML link to enable drill-down. Robert Allison of SAS informed me of a way to incorporate "tool tips" into my SAS GRAPH HTML maps as well as the drill-down capacity. This information and much, much more! are available on the SAS Web Site in the Technical Tips page.

You have already seen the creation of the hard-coded HTML link (LINKME) in the section on building presentation data sets above. To create a "tool tip", or an image that displays when you hover or roll over a HTML image with your mouse in Internet Explorer, use the following syntax:

```
altvar='ALT="State Code: ' || trim(left(state)) || '0D'x ||
'State name: ' || trim(left(fipnamel(state))) || '"';
```

The '0D'x hex value creates a line break in the text string. In this case, when you hover over the state of Massachusetts, you will see the following:

State Code: 25
State Name: Massachusetts

The example presented in SAS Technical Tips suggests that you COMBINE the drill-down and tool tips links into a single variable, which is an excellent idea. You can simply concatenate the two variables.

```
htmlvar=linkme|| ' ' || altvar;
```

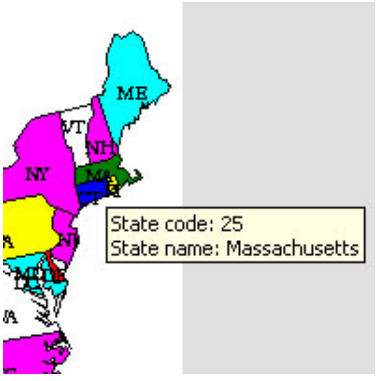
and

```
choro state / html=htmlvar;
```

instead of

```
choro state / html=linkme;
```

This will enable both drilldown and tool tips on your map.

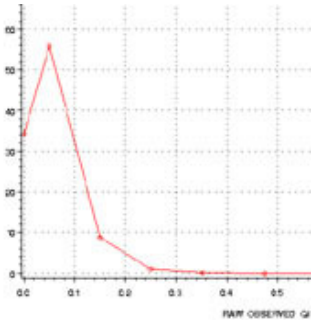


AN IMPORTANT TIP REGARDING SAS HTML GRAPHICS

It should be obvious from the examples above that creating a "clickable" graphic system can generate a LOT of HTML files. There are additional files that the SAS system creates when producing HTML graphics (HTML tables and reports are NOT affected by this). The SAS device used when producing HTML graphics such as maps and charts is the GIF device. SAS generates a separate GIF file for EACH TYPE of graphic, numbering multiple graphics of the same type sequentially. These GIF files MUST be in the directory where the HTML files reside and are called from using a browser. In addition, if you run more than one program generating the same types of graphics in the same directory, the GIF files from the most recent program will overwrite the older GIF files, creating errors. You can use the NAME= option in your graphics code to specify the names of your GIF files which eliminates this issue. Although it is not currently documented as available on all graphics procedures, it is.

Example:

```
ods html body="&qm.cht.htm";
goptions reset=global;
goptions device=gif border ftext=swissb;
symbol1 color=red interpol=join value=diamond;
pattern1 value=msolid color=red;
proc gplot data=ol_&qm;
    plot percent*ol_&qm / name="ol_&qm"
         vaxis=0 10 20 30 40 50 60 70 80 90 100
         haxis=0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1
         grid;
title3 "OL_&QM";
run;

ods html close;
quit;
```

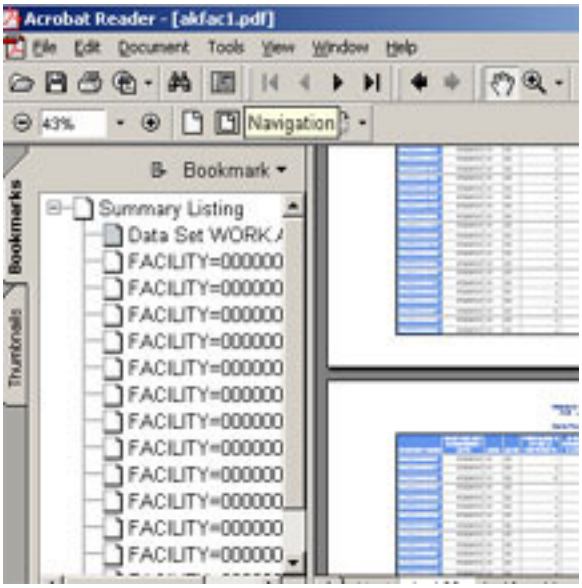


Note: Graphic is cropped to save room in the paper.

It is also possible to consolidate several graphic images into a single graphic file, cutting down on the number of files. This is particularly easy when using the PDF and HTML destinations. SAS allows you to combine output from multiple procedures when using the Output Delivery System. Both the PDF and HTML destination also allow you to index or bookmark the output from different procedures within a single PDF or HTML file. In addition, you can use by variables to dictate the "bookmarks" in your graphics image file.

Example:

```
options orientation=landscape papersize=letter leftmargin=.5
rightmargin=.5 topmargin=.5 bottommargin=.5;

ods pdf file="c:\tremor\akfac1.pdf"
author="Abt Associates Inc."
keywords="QI Resident Facility QIO"
subject="Quality Indicators"
title="Resident Level Quality Indicator Summary"
startpage=now
style=dd.facilweb;

ods proclabel="Summary Listing";
proc print data=akanal label uniform split='*';
var dateplug aa8a aa8b cres1 cpru2 cpru3 cinfx cbeh4 ccnt3
triggers;
id res_id;
by fac_itl;
pageby fac_itl;
format cres1 cpru2 cpru3 cinfx cbeh4 ccnt3 trigs.;
title1 'TREMOR / MEGAQI';
title2 "PDF - Alaska";
run;
quit;
ods pdf close;
```



Note: When you open a PDF file created using SAS Version 8 on any platform, you will get an error message telling you that the file is damaged and being repaired. The file will open correctly. SAS has fixed this problem in Version 9. Also, creating PDFs using Version 8 on the mainframe platform is difficult due to the large amount of work space needed (but it is possible!); another problem that has been addressed in Version 9.

MAINFRAME MAGIC

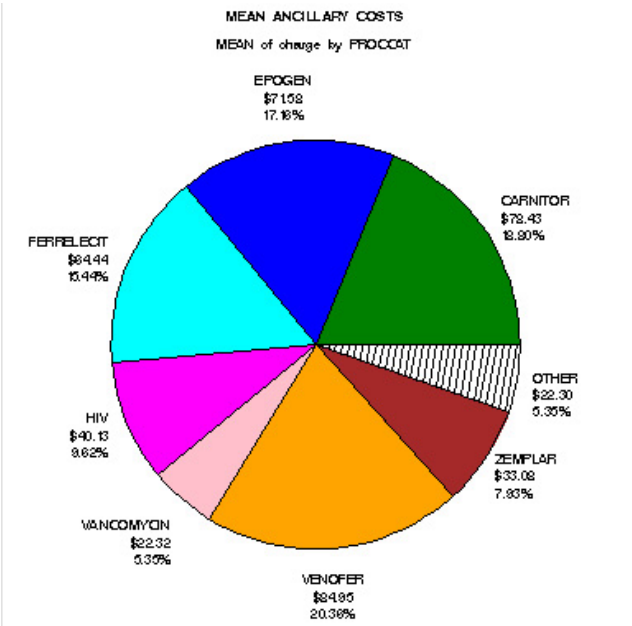
Production of HTML graphics and other SAS produced graphics is also possible on mainframe systems. There are a few quirks. The most important thing is that your life will be MUCH easier if you create a PDSE library to store your graphic files in. While you can create graphic files outside of a PDSE library, you have to create a separate file reference for each component of an HTML graphic file, for example (BODY, INDEX, CONTENTS, etc.) Having a library to store the images in enables you to use one command instead of many. Detailed information on ODS graphics on OS/390 systems can be found on the SAS website at www.sas.com/service/admin/mainframe/os390/tips.

WHAT IF I'M AFRAID OF MICE?

"Clickable" HTML documents, graphics or maps are very cool. However, there may be times when interactive HTML is not appropriate. For example, you may wish to publish your data in the form of a PDF document, or a data user may wish to manipulate the appearance of a graphic themselves without actually performing any SAS programming. You might want to insert SAS-created HTML documents or graphics into other forms of software such as Microsoft Word or Excel. SAS version 8 and the Output Delivery System allow for such desires with such output formats as JAVA, Active-X, XML, PDF, etc. as well as HTML, SAS listings, and data sets. Below follows four pie charts created using the SAS system on the same data using different devices. Portions of the code used to create each pie chart precede the images.

STATIC HTML PIE CHART:

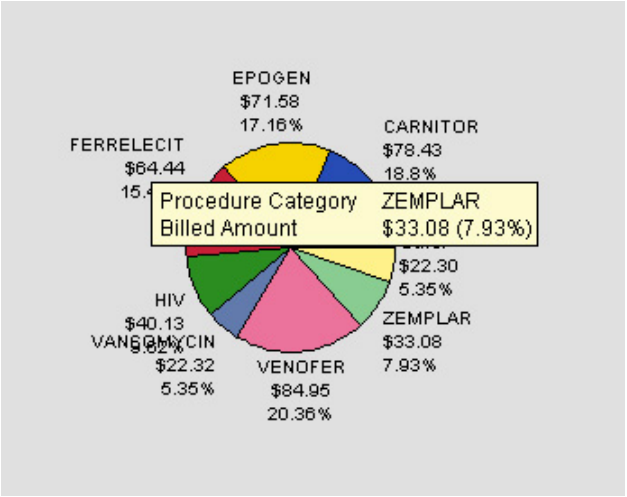
```
options ps=50 ls=80 errorabend;

libname ee '.';
title1 'ESRDPAC Bundle Analysis';
run;

/* specify the directory to write HTML files to */

filename odsout '.';

ods listing close;
goptions reset=global;

goptions device=gif hsize=4 in vsize=4 in ftext=swiss;

ods html body='daypie1.htm' path=odsout;

proc gchart data=ee.daylevel (where=(proccat in
    ('EPOGEN','FERRELECIT','IRON_DEXTRAN','HIV','CALCIJEX',
     'CARNITOR','VENOFER','VANCOMYCIN','ZEMPLAR','OTHER')));
    pie proccat / sumvar=charge type=mean
                  percent=outside value=outside;
title1 f=swissb h=1 "MEAN ANCILLARY COSTS";
run;

quit;

ods html close;
```



JAVA PIE CHART:

JAVA graphics can be manipulated by right clicking on the image with the mouse. Rolling over portions of the graphics gives additional information as portrayed below. You can change the graph type, colors, etc. with a single click. *Note: You MUST have SAS Version 8 or above OR the special JAVA applet installed on the displaying system or the JAVA capabilities will not work.*

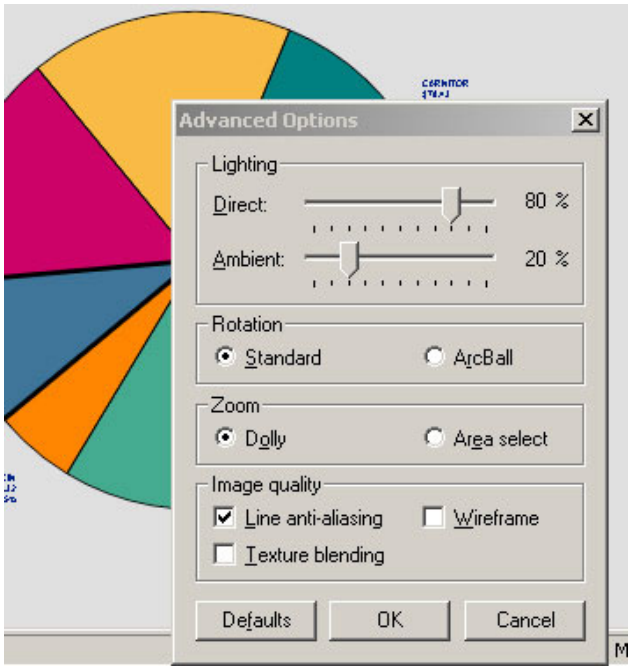
```
goptions reset=global;

goptions device=java hsize=4 in vsize=4 in ftext=swiss;

ods html body='javapie1.htm' path=odsout;

proc gchart data=ee.daylevel (where=(proccat in
    ('EPOGEN','FERRELECIT','IRON_DEXTRAN','HIV','CALCIJEX',
     'CARNITOR','VENOFER','VANCOMYCIN','ZEMPLAR','OTHER')));
    pie proccat / sumvar=charge type=mean
                  percent=outside value=outside;
title1 f=swissb h=1 "MEAN ANCILLARY COSTS";
run;

quit;

ods html close;
```



ACTIVE-X PIE CHART:

Active-X graphics can also be manipulated by right clicking on the image with the mouse. The capabilities are slightly different from those of JAVA graphics. *Note: You MUST have SAS Version 8 or above OR the Active-X application downloaded from Microsoft installed on the displaying system or the Active-X capabilities will not work.*

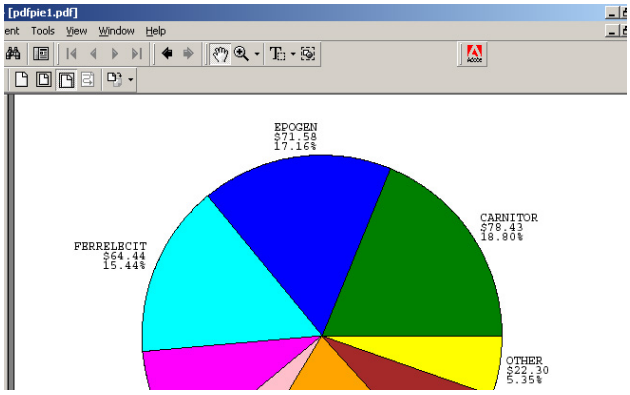
```
goptions device=activex hsize=4 in vsize=4 in ftext=swiss
htext=.5;

ods html body='actxpie1.htm' path=odsout;

proc gchart data=ee.daylevel (where=(proccat in
    ('EPOGEN','FERRELECIT','IRON_DEXTRAN','HIV','CALCIJEX',
     'CARNITOR','VENOFER','VANCOMYCIN','ZEMPLAR','OTHER')));
    pie proccat / sumvar=charge type=mean
                  percent=outside value=outside;
title1 f=swissb h=1 "MEAN ANCILLARY COSTS";
run;

ods html close;

quit;
```



PDF PIE CHART:

SAS produced PDF graphics are not interactive in any way. There are, however, advantages to producing large volumes of graphics and/or tables using the SAS V8 PDF device, as you can manipulate bookmarks and styles to make accessing the different graphics very easy. It also makes it hard to "manipulate the data", a very credible fear of web publishers whether using the SAS system or not!

```
goptions reset=global;

ods pdf file='c:\esrdpac\pdfpie1.pdf';

proc gchart data=ee.daylevel (where=(proccat in
    ('EPOGEN','FERRELECIT','IRON_DEXTRAN','HIV','CALCIJEX',
     'CARNITOR','VENOFER','VANCOMYCIN','ZEMPLAR','OTHER')));
    pie proccat / sumvar=charge type=mean
                  percent=outside value=outside;
title1 f=swissb h=1 "MEAN ANCILLARY COSTS";
run;

quit;

ods pdf close;
```



Note: Graphic cropped to save room in paper.

CONCLUSION

SAS Version 8 provides many new, and some improved older tools, to produce some truly amazing results. Intranet or internet publishing in a variety of useful forms becomes easy using SAS Version 8's Output Delivery System. In particular, attractive and useful interactive graphic presentation systems such as "clickable" maps can be created using SAS Version 8's capability to produce custom HTML graphics. It is even possible, with some clever coding, to animate the graphics, all completely within the SAS system. Presentation of data using the SAS system has come a long way since the ubiquitous PROC PRINT output of the 70's, carrying the SAS system into the new millennium in STYLE!

REFERENCES

SAS Online Documentation (PC SAS V612, PC SAS V8, AIX UNIX SAS V612, AIX UNIX SAS V8, RED HAT LINUX 6.2 SAS V8)

The Complete Guide to the SAS Output Delivery System, Versions 7-1 and 8

ACKNOWLEDGMENTS

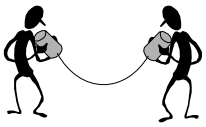
The author wishes to acknowledge Ray Pass and Dana Rafiee, who inspired the "clickable" map with their tutorials on adding hot-link drill-down capability to HTML output and Web Publishing, Patrick McGown for his illuminating paper on publishing Adobe PDFs using SAS, and Mike Zdeb, the guru of SAS GRAPH maps. In addition, SAS Technical Support has been incredibly helpful, especially Robert Allison. If you haven't checked out SAS Technical Tips (www.sas.com/service/techtips) yet, you are in for a pleasant surprise!

CONTACT INFORMATION

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

Louise Hadden
Abt Associates Inc.
55 Wheeler St.
Cambridge, MA 02138
Work Phone: 617-349-2385
Fax: 617-349-2675

Email: louise_hadden@abtassoc.com



KEYWORDS

SAS; UNIX; LINUX; ODS; HTML; ACTIVE X; PDF; GIF; JAVA; SAS GRAPH; MAPS