GIS Capabilities in Old vs. New SAS® Software

Richard O. Smith, Data Explorations, Carlsbad, CA
Arthur L. Carpenter, Data Explorations, Carlsbad, CA

ABSTRACT

Fifteen years ago, the authors came up with methods to determine if kelp bed sampling points were located within the boundaries of substrate boundaries. The data were then analyzed and plotted on a map of the study areas. Mapping and the ability to determine relationships between data and geographic locations or features were very limited in SAS software in the mid 80's.

The author's methods were the precursors of GIS at the time and worked well within the SAS software system. This 'Old' style of GIS will be compared to the capabilities present today in SAS/GIS®, and the new SAS® Bridge for ESRI.

KEYWORDS

GIS, SAS/GIS, SAS/GRAPH, POLYGON

INTRODUCTION

During the 1980's and early 1990's the authors were involved in a long-term study to determine the effects of the warm-water effluent from the San Onofre power plant in southern California, on the offshore kelp forests nearby. Many aspects of the ecology of the kelp forests were examined over many years, resulting in large amounts of data collected at specific locations within and near the kelp forests.

It was important that the scientists were able to examine results of field surveys rapidly as they related to locations sampled. Software was written to present collected data and summarized information in the form of maps of the study areas. Base maps were created for each region and overlaid with a wide variety of chemical and biological information.

Since GIS software (Geographical Information System) is a recent phenomenon, the methods used were the author's early versions of GIS. Software had to be written to handle a wide variety of factors. We examine several methods that simulate GIS functions that are often used in GIS systems today.

During the long-term study, bottom substrate boundaries, and kelp density were determined using side-scan and down-looking sonar. The substrate boundaries were digitized as x,y coordinates. The kelp density was collected at specific locations collected as x,y coordinates. The substrate boundaries were often very complex polygons, making it difficult to determine areas. It was also important to identify kelp density points that were within the substrate boundaries, and often important to identify and separate kelp density sample points that were near the borders of the substrate boundaries.

We will examine how we used SAS software to 1) calculate the areas of substrate, 2) to identify kelp density data points that lie within the substrate boundaries, and 3) to identify whether the points are within a selected distance from boundaries.

The methods developed using the older versions of SAS software will be compared to the capabilities available today in the new versions of SAS software.

METHODS USING OLDER SAS VERSIONS

During the mid-eighties the authors used SAS software on a CMS mainframe system. The mainframe system filled a large, refrigerated, 'state-of-the art' computer room with computing power much less than the normal desktop found today. However, the older SAS software contained powerful capabilities that are still good today.

Base SAS software contained all of the necessary capabilities needed to accomplish the three tasks above. Algorithms were created using basic data steps and the use of arrays. The SAS Macro Language added the ability to easily rerun the code for multiple surveys without extensive recoding.

Detailed descriptions of the concepts used to determine areas, and to detect if points are inside of polygons were previously discussed by Smith and Carpenter (1989). We will briefly examine the methods used and the additional method to detect points within a specific distance of polygon boundaries.

A SAS program was created to perform all three of the listed tasks and create permanent SAS datasets. This program was condensed for publication and is provided at the end of this paper (Table 1). Using the new datasets, a SAS/Graph® program was then used to create the example plot (Figure 1), seen below. The plot program is not included, but is a fairly simple program that plots the Kelp Density shot-points by location (Y*X=location) and is annotated with the substrate boundaries.

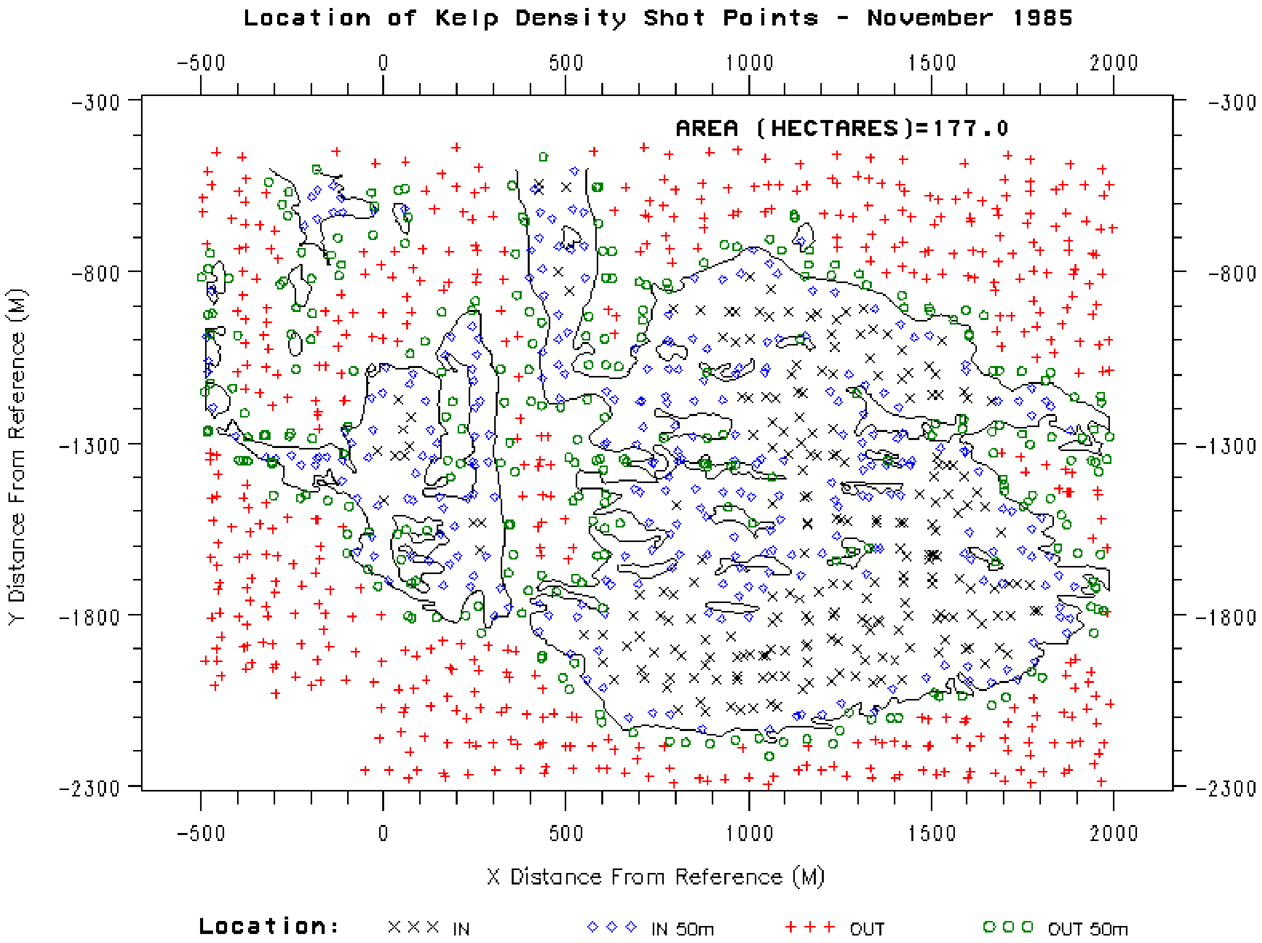


Figure 1. Plot of San Onofre kelpbed substrate borders with kelp density sonar shot-points displayed by location as relative to the substrate boundaries. Shot-points are further identified within a 50m buffer zone.

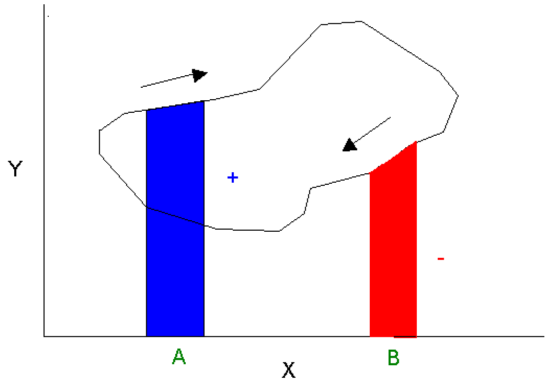
Methodology – Determining the Area

The area of a polygon is determined by calculating the area of a series of individual trapezoids. Each two successive pairs of coordinates define a trapezoid based on the x-axis. Figure 2 shows that areas of trapezoids along the top (A) of the polygon are positive and trapezoid areas along the bottom (B) are negative. The area of a trapezoid may include area that is either interior or exterior to the polygon or both. For every trapezoid containing both interior and exterior areas, there is another that contains the exterior area only. The difference between the two is the interior area.

The Polygon area is simply the summation of the areas defined by successive trapezoids. Since the area will have a different sign depending on the direction of increasing or decreasing x, the sum of the areas automatically performs the subtraction of exterior areas. The area of a polygon whose x,y coordinates are ordered in a clockwise manner around the border of the polygon will be positive. The area will be negative for polygons whose x,y coordinates are ordered counter-clockwise around its border. Therefore, it is necessary to take the absolute value of the summation to determine the polygon area.

This technique is valid for polygons where portions of the border pass below the x-axis. Passing to the left of the y-axis has no effect. Polygons of empty space that lie inside the interior of larger polygons are identified by a variable (loc) with a value of –1. All polygon areas are multiplied by the loc variable before calculation of total area. In Figure 1, the area is placed on the map using annotate.

Figure 2

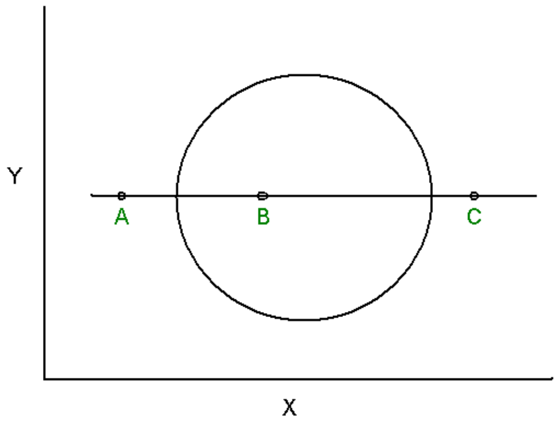


Methodology – Determining Interior Points

The sequential string of x,y pairs that form the border of the polygon contain information only on the location of the border. It is possible however, to determine if a single x,y point that is not on the border, is in the interior of the polygon. The method used is based on a series of determinations at constant values of the y variable. If a line with constant y cuts a polygon, the line will intersect the border at one or more locations. If the x value of each data point in the related data set is compared at each y to the x values of the points of intersection from the polygon data set, it is possible to determine which are interior points.

In Figure 3, point A must be outside of the polygon because it has an x value that is less than both of the boundary points. By the same token, C is also outside because its value of x is larger than both boundary points. B is the only interior point because its x lies between the two boundary points. This concept, *interior points can be defined by pairs of boundary points*, is called the 'Two Points Rule' (TPR) and is the cornerstone of this method.

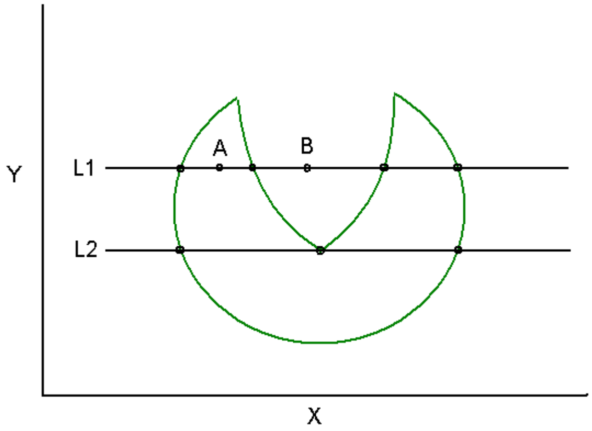
Figure 3



There are two exceptions to the Two Points Rule. Figure 4 shows a slightly more difficult polygon cut by two lines. The first (L1) cuts the polygon four times and has an imbedded exterior space. L1 follows the TPR if the points are taken in order as pairs. Point A is within the first pair and point B is outside of both pairs. L2 intersects the polygon three times with the center point at a local minimum. This makes it difficult to form the pairs needed to satisfy the TPR. Pairs can be formed if all local maximum and minimum points are duplicated. This allows line L2 to behave like L1 by creating an exterior zone of zero width and thus satisfying the TPR.

The second exception to the TPR occurs when polygon borders contain a series of constant y values. To solve this problem, every second data point along the constant y border is offset either above or below the y-axis by 0.5 units of resolution and any newly created maxima or minima are duplicated as above.

Figure 4



It is important to determine the resolution needed by the subject data and to modify the polygon border datasets to this resolution using interpolation. If this is not done, there will be missing y values on the polygon border to compare against the subject data. In the Figure 1 example, it was determined that the kelp density points were accurate to +/- 3 meters. A resolution of 1 meter was chosen and y was rounded to 1 meter in both the substrate border data set and the kelp density data set.

Methodology – Buffer Zones

Creating buffer zones is a fairly simple task. The polygon dataset is read into a data set and macro variables are created for each x and y value, and the total number of polygon points are counted and a macro variable is used to also capture this value. The subject data (e.g., Kelp Density) is then read into a data set and a macro loop is used to detect if any of the polygon points are less than or equal to the selected buffer distance. Points that are within the selected distance are flagged. In the Figure 1 example, this step occurred after first determining if points were interior or exterior to polygons. The IN or OUT location of points were then modified to become IN, IN 50m, OUT, or OUT 50m which included the selected buffer distance.

The sequence used to create the final KDENS data set used in Figure 1 is as follows: 1) Resolution needed is determined and y values rounded to resolution, 2) Polygons are completed by duplicating the first data point at the end of the polygon x,y series, 3) Area is calculated, 4) Constant y values are adjusted and local maxima and minima are doubled, 5) New data points are created by interpolation to the defined resolution, 6) An array of x intersect values is loaded for each y, and these values of x are compared against the values stored in the array using the TPR to determine if the point is interior or exterior to the polygon, 7) Kelp density data points are compared against each polygon point to determine if it is within the selected buffer zone distance, 8) A permanent data set is created containing the original kelp density points which now also contains an Area variable and a Location variable. Location identifies if points are interior to polygons, and also if within buffer zone.

METHODS USING OLDER SAS VERSIONS

Newer SAS/GRAPH Options and ODS

Newer versions of SAS, SAS/GRAPH, and the inception of ODS have given a boost to the capabilities of these basic components of SAS. Using SAS/GRAPH and ODS html with drill-down capabilities, we can create a more powerful presentation of the data. Newer graphics devices allow the creation of a number of file forms that are compatible with presentation on the web.

The macro language can also be used to generalize the ODS statements so that the routing of the graphs can be more easily controlled.

SAS/GIS

SAS/GIS is an interactive GIS system within the SAS system that first became available with version 6.12. The early version was geared toward the business industry and the limitations caused the author to begin using ARC/INFO® and ArcView® for mapping purposes in the environmental/ecological field. The newer version contains many features that make it more useful for our purposes.

SAS/GIS, like other GIS systems, use two types of data: Spatial Data and Attribute Data. Spatial data contains the coordinate and identifying information that creates the map. It can be in the form of points, lines, or areas (polygons). Attribute data are information variables that can be linked to the spatial data. This system has the capability to import GIS information from a wide variety of systems, most importantly (for the authors), ESRI's ARC/INFO.

SAS/GIS allows one to select features from a map, and then perform actions on the attribute data. Of most importance to our study, was the ability to 1) display observations that relate to selected features, 2) the ability to subset attribute data that relate to selected features, and 3) perform analyses of subset data using SAS programs. There are many other actions that this GIS software can perform.

The above concepts and additional information on SAS/GIS can be found on the SAS website. The following links were useful for this paper:
1) Product Information on SAS/GIS:
http://www.sas.com/products/gis/
2) SAS/GIS Tutorial:
http://support.sas.com/training/elearn/tutorials/v8/gis/main_spl.htm
3) SAS/GIS Example Programs:
http://support.sas.com/techsup/sample/sasgis_samples.html

For the study discussed in the first half of the paper, a map is created using substrate boundary data to create thematic layers that identify different substrate types. Additionally the substrate boundaries were used to create line features in a separate layer. Kelp density sonar survey points were used to create a point layer. Points that lie within the boundaries of the substrate layer can be identified by first creating a lattice of hierarchy of areas in the database, display the kelp density points, make them selectable, select all points, edit the point layer information and edit Locals to Set Area Attributes. The point layer now contains a value assigned that identifies the polygon. A SAS dataset can now be created by selecting Actions menu or by editing the Attribute Data Sets window.

Although there is no room for examples in the paper, the poster for this paper should include an example.

SAS® Bridge for ESRI

The new SAS® Bridge for ESRI gives users of ARC/INFO use of the analysis power of the SAS system. Data remain in the ARC/INFO system, but provides direct links to SAS to performs analyses of data and present them directly in ARC/INFO.

For those that use ARC/INFO, this provides the power of SAS software to perform analyses not found in the GIS system.

SUMMARY

It is possible to use the DATA step and macro language statements to estimate the area of irregular polygons, to determine whether or not any given point resides within the polygon, and it is even possible to determine which points are close to the border of the polygon.

Recent advances in SAS/GRAPH and ODS allow the enhanced graphical presentation of the polygon and its interior and exterior points.

Similar determinations of polygon area and interior/exterior point assignments can now also be made using SAS/GIS. The new SAS® Bridge for ESRI provides ARC/INFO users links to SAS software to perform analyses within the ESRI system.

Table 1. Code used to calculate areas, identify points within polygons and buffer zones. Compressed 2 pgs.

```
* dbkdens.sas
  01Aug2003 - Richard Smith & Art Carpenter.
  Calculate areas of substrate polygons, ID
  kelp density points within substrate borders,
  identify points lying within selected buffer
  distance from polygon borders. Edited from
  programs developed by Smith/Carpenter in 1988.
*--------------------------------------------*;

%macro dbkdens(survey,date,dbson,dbsub,dbkdens,
         dbout=,bufferflag=1,bufferdist=50,
         printflag=0,no=no);
options dquote &no.macrogen &no.symbolgen
&no.mprint notext82 nocenter;
%let path = c:\projects\mrc;
%let wuss = c:\usergroups\wuss2003;
libname dbkdens  v604 "&path\dbkdens";
libname dbanno   v604 "&path\dbanno";
libname dbsonden v604
"&path\mrcdbs\physchem\dbsonden";
libname wuss "&wuss\poster\sasdbs";
%if &dbout =  %then %let dbout = &dbkdens;

* data set from which densities are selected ;
data dens1;  set dbsonden.&dbson;
  x=xloc;y=yloc;
  y=round(y,1); /* rounds to nearest interger */
  db="&dbsub";  bed=substr(db,1,3);
  if bed='sok' then do;
     if x>=-500 & x<=2000 & y<=-500 & y>=-2500;
  end;
  if bed='smk' then do;
     if x>=-5500 & x<=-3500 & y<=0 & y>=-1500;
  end;
  keep y x kdens;  run;
proc sort data=dens1;  by y;  run;

* calculate area of kelpbeds *;
data area1;  set dbanno.&dbsub;  by polygon;
  retain xo yo ox oy;
  if x=. or y=. then delete;
  if first.polygon then do;
     ox=x; oy=y; xo=.; yo=.;
  end;
  area = ((yo+y)/2)*(xo-x);  output;
  xo=x; yo=y;

  * Use 1st obs in polygon as last added point
    to complete unfinished polygons;
  if last.polygon then do;
     area = ((y+oy)/2)*(x-ox);  output;
  end;
  run;

*** sums total area for each polygon ***;
proc means data=area1 noprint;
  by polygon;  var area;  id loc;
  output out=area2 sum=area;
  run;

* change neg. polygon areas to pos., multiply
  times loc(1 or -1) to subtract inner polygons,
  and sum polygons *;
data area3;  length survey 3.;  set area2;
  survey=&survey; date=&date;
  area=abs(area)*loc; format date date7.;
  run;
proc means data=area3 noprint;  by survey date;
  var area;  output out=area sum=area;
  run;

* identify kdens values in & out of bed borders;
```

```
* create final pt equal to 1st so poly complete;
data poly1;  set dbanno.&dbsub;  by polygon;
  y=round(y,1); /* rounds to nearest interger */
  retain firstx 0 firsty 0;
  if first.polygon then do;
     firstx = x; firsty=y;
  end;
  output;
  if last.polygon then do;
     x=firstx;  y=firsty;  output;
  end;
  keep polygon x y;     run;

* adjust y values so that there are no
  sequential data with same y values *;
data polyh;  set poly1;  by polygon;
  retain ld d oy ox oy2 ty tx;
  if first.polygon then do;
     ld=.; d=.; oy=.; ox=.; oy2=.; ty=.; tx=.;
  end;
  ld=oy-oy2;  d=y-oy;  ty=y; tx=x;
  if ld^=0 and d^=. then do;
     y=oy;  x=ox;  output;
     y=ty; x=tx;
  end;
  if ld=0 then do;
     if d=0 then do;
        y=oy+0.5;  x=ox;  output;
        oy=oy+0.5;  y=ty;  x=tx;
     end;
     if abs(d) > 1 then do;
        y=oy + sign(d); x=ox;  output;
        oy=oy + sign(d);  y=ty;  x=tx;
     end;
     if abs(d) = 1 then do;
        y=oy - sign(d); x=ox;  output;
        oy=oy - sign(d);  y=ty;  x=tx;
     end;
  end;
  if last.polygon then output;
  oy2=oy;  oy=y;  ox=x;
  keep polygon x y;     run;

* create a value for each y by interpolation *;
data poly2;  set polyh;  by polygon;
  retain oldy oldx;
  if first.polygon then do;
     oldy=.; oldx=.;
  end;
  * eliminate exact duplicates;
  if x=oldx and y=oldy then delete;
  diffy=y-oldy;  diffx=x-oldx;  i=0;
  if abs(diffy) >1 then do
    i=sign(diffy) to diffy by sign(diffy);
     y=oldy+i;  x=oldx+((diffx/diffy)*i);
     output;
  end;
  else output;
  oldy=y; oldx=x;  keep polygon x y;  run;

* create dup y value when vert direction change;
data poly3;  set poly2;  by polygon;
  retain ld d oy ox oy2 ty tx fd n;
  if first.polygon then do;
    ld=.;d=.;oy=.;ox=.;oy2=.;ty=.;tx=.;fd=.;n=0;
  end;
  n=n+1;  id=sign(oy-oy2);  d=sign(y-oy);
  if n=2 then fd=d;  ty=y; tx=x;
  if (ld ^= d and ld ^=.) then do;
     y=oy;  x=ox;  output;
     y=ty;  x=tx;
  end;
  if last.polygon and fd = d then delete;
  output;
  oy2=oy;  oy=y;  ox=x;  keep polygon x y;  run;
```

```
proc sort data=poly3 out=poly3y;  by y x;  run;

* calculate the number of x per y *;
proc means data=poly3y noprint;
  by y;  var x;  output out=mnxdata n=nx;  run;

* transpose x up for each y (eg. y x1 x2 x3 x4);
proc transpose data=poly3y out=tpdata
  prefix=x (drop=_name_);  by y;  var x;  run;

* merge tposed poly dat & mnxdata w/kdens data;
data kdens;
  merge dens1(in=in1) mnxdata tpdata(in=in2);
  by y;
  * create dummy var used to dimension array;
  extrax = .;
  * selects only y's present in subject data *;
  if ^in1 then delete;
  * label data in/out of polygon differently *;
  i=0;  good=0;
  if nx ^=. then do;
     array xx {*} x1 -- extrax;
     do i=2 to nx by 2 until (good);
        if xx{i-1} <= x <= xx{i} then good=1;
     end;
  end;
  if nx=. then good=.;  length location $8;
  if good then location='IN ';
  else location='OUT' ;
  drop extrax i good;
  run;

* Identify points within a buffer zone;
%if &bufferflag = 1 %then %do;
   * create macro vars for substrate x,y points;
   data buffer;  set poly3;
     length ii $12;  i+1;  ii=compress(i);
     call symput('px'||ii,compress(x));
     call symput('py'||ii,compress(y));
     call symput('npoly',ii);
     run;

   * determine kdens pts within buffer distance;
   data kdens;  set kdens;
     inbuffer = 0; /* default inbuffer var */
     %do i = 1 %to &npoly;
       pxdist = x - &&px&i; pydist = y - &&py&i;
       distance = sqrt((pxdist*pxdist) +
                        (pydist*pydist));
       if distance<=&bufferdist then inbuffer=1;
     %end;

     * redefine the locations;
     if location = 'IN'  and inbuffer = 1
       then location = "IN &bufferdist.m";
     if location = 'OUT' and inbuffer = 1
       then location = "OUT &bufferdist.m";
     run;
%end;

* merge area of kelpbed with kdens data *;
data wuss.&dbout;
  if _n_=1 then set area;  set kdens;  run;

%if &printflag = 1 %then %do;
   proc sort; by location; run;
   proc print;
     title "Data Present in DBKDENS.&dbkdens";
  run;
%end;
%mend dbkdens; ********* END OF MACRO *********;

* run macro with 50m buffer for 1985 survey 7 *;
 %dbkdens(7,'19Nov85'd,SUR007,SOKSUB22,SOKSUR07,
          bufferflag=1,bufferdist=50) *;
```

REFERENCES

Smith, Richard O., and Arthur L. Carpenter. 1989. “Polygons: Calculating Area and Identifying Interior Points”. Proceedings of the 14th Annual SAS® Users Group International Conference, Cary, NC: SAS Institute Inc., pages 1182-1187.

ABOUT THE AUTHORS

Richard Smith and Art Carpenter are senior partners at Data Explorations. Data Explorations, a SAS Alliance Affiliate Member™, provides data management, analyses, and SAS programming services nationwide.

Richard O. Smith

Richard Smith has a Masters in Biology/Ecology and has provided complete data management and analysis services for numerous environmental research projects as a senior biologist, SAS programmer, and project manager. He has also provided programming and management services for the health related industries. He has been using SAS extensively since 1981 and is a SAS Certified Professional™.

Arthur L. Carpenter

Art Carpenter is a SAS Certified Professional™. His publications list includes three books on SAS topics (*Annotate: Simply the Basics*, *Quick Results with SAS/GRAPH® Software*, and *Carpenter's Complete Guide to the SAS® Macro Language*), two chapters in *Reporting from the Field*, and numerous papers and posters presented at various user group conferences. Art has been using SAS since 1976 and has served in a variety of positions in user groups at the local, regional, and national level.

AUTHOR CONTACTS

Data Explorations
2270 Camino Vida Roble, Suite L
Carlsbad, CA 92009

Richard O. Smith
(760) 438-1336
RSmith@SciX.com

Arthur L. Carpenter
(760) 945-0613
Art@Caloxy.com

TRADEMARK INFORMATION