

# Together at Last: Spatial Analysis and SAS® Mapping

Darrell Massengill, Alexander Kolovos, SAS Institute Inc., Cary, NC

## ABSTRACT

Spatial analysis and maps are a perfect match. Spatial analysis adds intelligence to your maps; maps provide context for your spatial analysis. The geostatistical tools in SAS/STAT® software can model and predict a variety of spatial data. SAS Mapping facilities enable you to create rich visualizations from that material. This presentation introduces a new framework that combines SAS spatial analytics with SAS mapping. Examples demonstrate how you can use the SAS/GRAPH® Annotate facility with the transparency specification (new in SAS® 9.3) to combine a predicted spatial surface with traditional SAS/GRAPH® maps, and show how to tap into the additional mapping resources of Esri software through the SAS® Bridge for ESRI. These tools empower you to make more intelligent maps and more informative spatial analyses.

## INTRODUCTION

Spatial analysis is mainly concerned with modeling and estimating processes and phenomena that live on the surface of the earth. Maps can depict such phenomena, give them context, and point out where they are in a practical sense. For example, a spatial analysis can tell you the longitude and latitude of sites or regions with the best access to wind or sun for alternative energy production, but a map tells you whether those sites are near cities or convenient roads.

SAS/STAT software provides excellent tools for modeling and visualizing spatial phenomena, and SAS/GRAPH gives you a great variety of convenient maps. This paper reviews these tools and their use, and presents the key to combining them—the SAS/GRAPH Annotate facility's transparency specification, which is new in SAS 9.3. These features are discussed within the context of a specific example, studying oil production in the Gulf of Mexico.

The first section of the paper presents the oil production data and steps through its analysis, leading to the plotted surface, shown in Figure 5, of predicted oil production over a particular rectangular region in the Gulf of Mexico. The remainder of the paper shows how to combine this analysis with map data for the Gulf. A brief conclusion discusses alternative mapping technology that enables you to present SAS/STAT spatial analyses, and points to future mapping capabilities for SAS software. An appendix provides Annotate macros that ease the task of overlaying a spatial analysis onto map data.

## SAS/STAT SPATIAL PROCEDURES AND SAS MAPPING

SAS/STAT has a suite of spatial procedures for focused analysis of point-referenced data (Kolovos 2010).

This type of data consists of observations for one or more attributes at a series of locations, and one typically looks to interpolate the attribute at unsampled locations. For the observations of each individual attribute, you can explore and model underlying correlations with the VARIOGRAM procedure, interpolate attribute values at unsampled locations with the KRIGE2D procedure, and simulate the mechanism that produced the attribute observations with the SIM2D procedure. This section explores how to integrate spatial analytics output that you acquire through these procedures into the SAS Mapping facilities. A complete example across the section guides you through the necessary steps.

Assume you are a researcher in the oil and gas business, and you are interested in estimates of oil yield across a specific location over a time period. Specifically, you are assigned to study the 2009 aggregate oil production, measured in barrels per month (BPM), in a selected rectangular area in the Gulf of Mexico. The area is defined by the coordinate intervals [-94, -87.7] longitude and [27.1, 29.8] latitude, which are measured in decimal degrees (DD). Your input data are nonmissing BPM observations from 4655 boreholes in the selected area. The oil yield raw data come from the Minerals Management Service (MMS) Web site that has a record of 10 years of monthly oil and gas production data. See the Bureau of Ocean Energy Management, Regulations and Enforcement Web site at <http://www.boemre.gov>. For this example, assume that you have initially processed the raw data to extract the total 2009 oil yield for each borehole; then, you have averaged the values in time to derive the BPM observation that corresponds to each borehole location in 2009.

In the following subsections, you see in detail how the SAS/STAT spatial procedures help you obtain analytical results for the BPM attribute across the area of interest. All example code and data are available at <http://support.sas.com/rnd/papers>.

## EXPLORATORY ANALYSIS

You start with the oil yield data that are stored in the data file "oil09.sas7bdat". The following DATA steps import the data into SAS and retain only the observations in the area of interest. For the purpose of the analysis in the following code, you need nonzero numbers in the data, so you represent zero BPM values with the small number  $1 \times 10^{-6}$ .

```

libname oilLib '.';
%let oil09temp=oilLib.oil09;

data oil09;
  set &oil09temp;
  if (bpm eq 0) then do;
    bpm = 1e-6;
  end;
  logbpm = log(bpm);
  if ((LonBot > -94) and (LonBot < -87.7) and
      (LatBot < 29.8) and (LatBot > 27.1)) then do;
    output;
  end;
  label logbpm='Log Barrels per Month';
run;

```

First, try to obtain an idea about the spatial distribution of the observations in the area and then proceed to study if and how they are correlated in space. Use the VARIOGRAM procedure to process the `bpm` variable in the `oil09` data set. The longitude of the BPM observations is in the `LonBot` variable, and their latitude is in the `LatBot` variable. Explore empirical correlation by placing observation pairs in distance bins and then measuring how dissimilar are the values of pairs in each bin. In effect, you specify parameters for the constant distance between lags and the maximum number of consecutive lags to consider. For the current example, the details of initial experimentation with these correlation parameters are not included. The findings from this step suggest that you can get a good feel of the empirical correlation with a lag distance value of 0.35 over the total length of 9 lags. Specify these numbers in the `LAGDISTANCE=` and `MAXLAG=` options, respectively, in the `COMPUTE` statement of PROC VARIOGRAM. Also, specify the `OBSERVATIONS` plot option to obtain a map of the BPM observations. Use the following statements:

```

ods graphics on;

proc variogram data=oil09 plots(only)=(observations equate);
  compute lagdistance=0.35 maxlag=9;
  coord xc=LonBot yc=LatBot;
  var bpm;
run;

```

Figure 1 shows the spatial distribution of the 4655 boreholes in the selected area in the northern Gulf of Mexico:

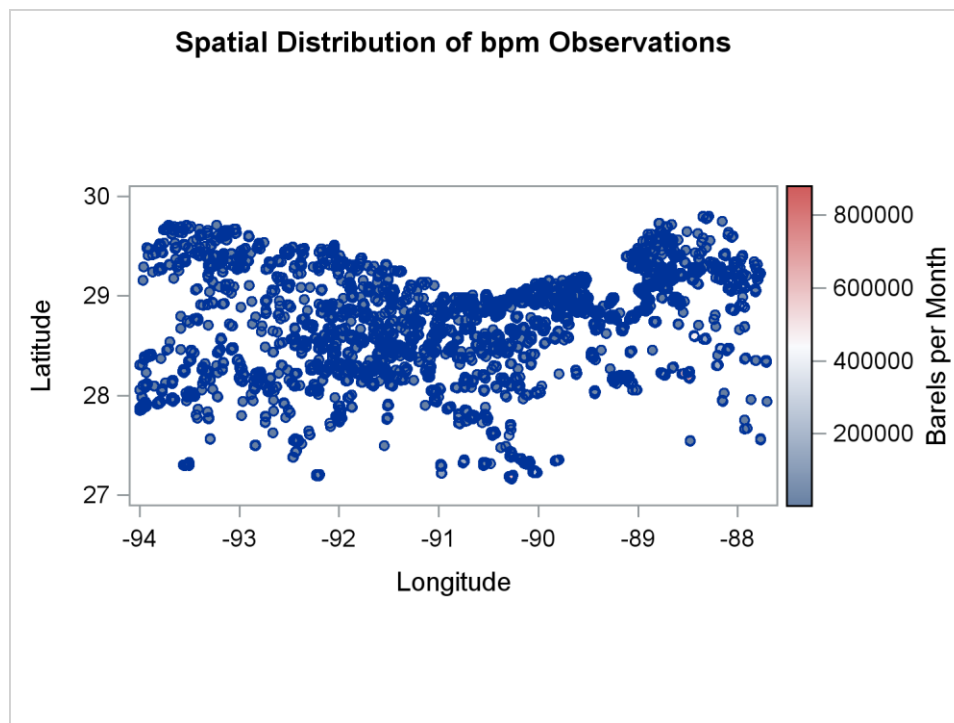


Figure 1. Spatial Distribution of the 4655 Boreholes in the Study

## CORRELATION ANALYSIS

Experience indicates that it is better to model BPM on the logarithmic scale, with the adjustment discussed previously for boreholes that yielded no oil at all. The following PROC VARIOGRAM statements compute a semivariogram model for  $\log(\text{BPM})$  and save it in the item store named `OilStore`. For the details of a VARIOGRAM analysis, see Kolovos (2010).

```
proc variogram data=oil09 plots(only)=(fit equate);
  store out=OilStore / label='Gulf Oil 2009 Correlation Model';
  compute lagdistance=0.35 maxlag=9;
  coord xc=LonBot yc=LatBot;
  var logbpm;
  model form=(exp);
  parms (28) (20) (0.25);
run;
```

## OIL YIELD PREDICTION

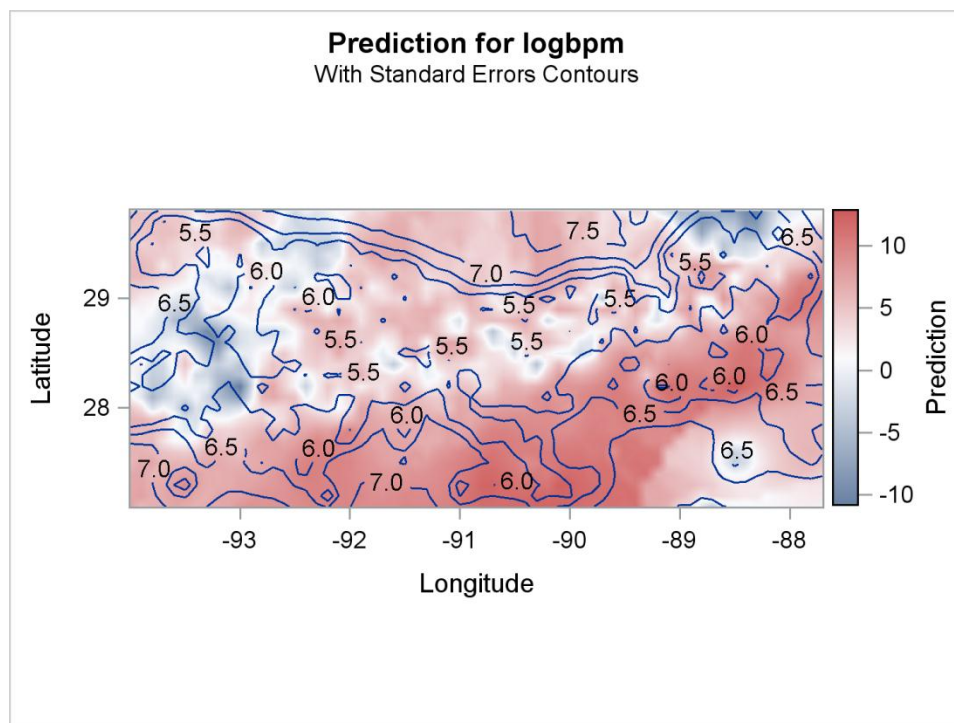
With the semivariogram model you have saved in the `OilStore` item store, you are ready to estimate oil production over your region of interest. A rectangular grid of 1792 nodes is used, where the inter-node distance is 0.1 degrees in the horizontal and vertical directions. Specify this grid in the GRID statement of PROC KRIGE2D. For the interpolation, consider up to the 50 closest neighbors within a radius of 1 degree from each prediction node. Specify the corresponding values into the MINPOINTS=, MAXPOINTS=, and RADIUS= options of the PREDICT statement. Kriging prediction requires a correlation model. By specifying the STORESELECT option in the MODEL statement, you request that PROC KRIGE2D uses model input from the input item store in the IN= option of the STORE statement. This is where you invoke the item store `OilStore` that you saved earlier in PROC VARIOGRAM. Finally, request the output data set `outOil` with the predicted  $\log\text{bpm}$  values in the OUTEST= option of the PROC statement, and a plot of the predictions on the requested grid with the PRED plot option. Specify the following statements:

```

proc krige2d data=oil09 outest=outOil
  plots(only)=(equate pred(fill=pred line=se));
  store in=OilStore;
  coord xc=LonBot yc=LatBot;
  grid x=-94 to -87.7 by 0.1 y=27.1 to 29.8 by 0.1;
  predict var=logbpm radius=1 minpoints=1 maxpoints=50;
  model storeselect;
run;

```

Figure 2 shows a surface of the predicted `logbpm` values. The contours indicate the prediction standard error. According to the predicted values, the highest yields are located farther off of the coast to the southeastern part of the domain, whereas the lowest average production is observed at the western part. This map provides a useful indication of the spatial distribution of oil yields in 2009. The prediction errors appear to be relatively high throughout the domain and, as you would expect, they increase as you move farther away from the observations.



**Figure 2. Prediction and Standard Error for logbpm**

## SAS MAPPING INTEGRATION

The SAS Mapping facilities enable you to produce maps that show the preceding results in the context of the land area that is nearby. The `bpmPlotData` data set that is generated below contains the output grid coordinates and the data for both the predicted BPM and `log(BPM)` values.

```

data bpmPlotData;
  set outOil;
  bpm = exp(Estimate);
  logbpm = Estimate;
  stdErrOrg = exp(StdErr);
  Longitude = gxc;
  Latitude = gyc;
run;

```

A number of mapping technologies can be used with SAS to show this data. This paper focuses on SAS/GRAPH mapping but also takes a brief look at SAS Bridge for ESRI. In addition, the paper shows how this example will look with future mapping technologies.

### SAS/GRAPH Mapping and Annotate Facilities

In order to make this example easy to use and easy to reuse with different data, several SAS Macro programs and SAS Macro variables are used. The Macro code is included with the example and in Appendix 1. The data set name and variable name of the grid area are set using macro variables. Using macro variables enables you to change data without rewriting the program. The following code sets the data set and variable names:

```
%let oildata=bpmPlotData;
%let predvar=logbpm;
/* %let predvar=bpm; */
```

First, create a heat map-like grid with the predictive values, using the SAS/GRAPH Annotate Facility. The bpmPlotData data set contains the log(BPM) and BPM values. Before creating the actual grid, you need to set the color for each grid cell based on the value of logbpm (or bpm). In order to make the heat map grid clearer, we use shades of green for negative values, shades of black closer to zero, and shades of red for positive values. The most negative numbers are the brightest green. The most positive numbers are the brightest red. The code can easily be changed to use other colors. The colors must be scaled by the maximum and minimum logbpm value and fit into a range of 0 to 255. You can find the maximum and minimum values by looking in the data set and setting the macro variables maxval and minval. The other macro variables were set in the preceding code.

```
%let maxval = 12.9;
%let minval = -10.8;

data grid;
  set &oildata;
  length hx $2;

  /*make negative numbers green to black */
  if (&predvar < 0) then do;
    cnum=(&predvar * (255/&minval)); /*scale colors */

    /*Make sure we stay within the threshold values*/
    if (cnum < 0) then cnum=0;
    else if (cnum > 255) then cnum=255;
    hx=put(cnum,hex2.); /*convert to hex*/
    /* Transparency for 9.3 and up*/
    color="RGBA00"||trim(left(hx))||"0099";
  end;

  /* Make Positive values black to Red */
  else do;
    cnum=(&predvar * (255/ &maxval));
    /*Make sure we stay within the threshold values*/
    if (cnum < 0) then cnum=0;
    else if (cnum > 255) then cnum=255;
    hx=put(cnum,hex2.);
    /* Transparency for 9.3 and up*/
    color="RGBA"||trim(left(hx))||"000099";
  end;
  x=longitude; y=latitude; /* use x and y values for GMAP */
run;
```

Transparency allows the land areas to be visible under the grid. The color values set in the preceding code use transparency, which was introduced in SAS 9.3. Transparent colors are represented with the letters RGBA followed by the hexadecimal values for red, green, and blue. These values are followed by the hexadecimal values for transparency. Each value can be between 0 and 255. So, RGBAFF00007F is bright red that is half transparent. For non-transparent colors, the colors start with CX and do not have the transparent color at the end. CXFF0000 is bright red. To make the code work with SAS 9.2 and earlier, you must replace the color setting code in the preceding code with the following lines of code:

```
color="CX00"||trim(left(hx))||"00";
```

```
color = "CX" || trim(left(hx)) || "0000";
```

Instead of looking in the data set for the min and max values, it is easier to automatically calculate them. SAS Macro code is provided to automatically calculate the values and set the macro variables. The first parameter is the data set name and the second is the variable to process. Replace the %let statements in the code sample on page 5 with the following code:

```
%include 'mpmacros.sas';
%get_minmax(&oildata,&predvar);
```

Now that we have assigned colors for each grid cell, we create an annotated grid by using the Annotate Facility. The `bpmPlotData` data set contains the coordinate point for the starting corner of each square grid cell. Each grid cell is the distance from one corner to the next one. This grid cell is 0.1 degrees in each direction.

Using a re-usable macro program to create the rectangular areas is much easier than learning to use Annotate. This macro is provided with the example code. Use the following code:

```
%make_rectangle(annogrid,grid,.1,.1,color,color,'Y','N',&predvar);
```

The parameters are:

- the output annotate data set
- the input data set
- the width of each rectangle
- the height of each rectangle
- the line color variable
- the fill color variable
- the specification of whether the rectangle is to be filled ("Y" or "N")
- the specification of whether the coordinate is the center point instead of the corner ("Y" or "N")
- a value for the tooltip

The `annogrid` data set now contains everything necessary to draw the grid on the map in the proper location and you are ready to prepare the map. Because this area is in the Gulf of Mexico, you draw this map on a US map. The grid data is in degrees, but the map is in radians. You can easily convert the map with a provided macro program:

```
%to_degrees(mystates, maps.states);
```

The map contains all 50 states, making the grid area quite small by comparison and more difficult to see. Only the Gulf region states are of interest, so subset the map to contain only Texas, Mississippi, Alabama, and Louisiana:

```
data mystates;
  set mystates;
  if ( (state eq sfips("TX"))
      or (state eq sfips("MS"))
      or (state eq sfips("AL"))
      or (state eq sfips("LA")) ) then do;
    output;
  end;
run;
```

This map looks good, but Texas is such a wide state that it still is a large part of the map area. One way to fix this is to split the state west of the southernmost point so that only the relevant coastline is shown:

```
data mystates;
  set mystates;
  if (state eq sfips("TX")) then do;
    /*stop it West of the southern-most point*/
    if (x > -98) then output;
  end;
  else output;
run;
```

All of the subsetting and splitting code in the preceding sample can be replaced with a call to PROC GPROJECT. This give you a similar view and makes a better looking map. In this case, you are not changing the projection of the map but using the clipping functionality instead. Note that PROJECT is set to NONE and four clipping boundaries are set with LONGMIN, LONGMAX, LATMIN, and LATMAX:

```
proc gproject data=mystates out=mystates dupok project=NONE
  longmin=-98 longmax=-84 latmin=25 latmax=35;
  id state;
run;
```

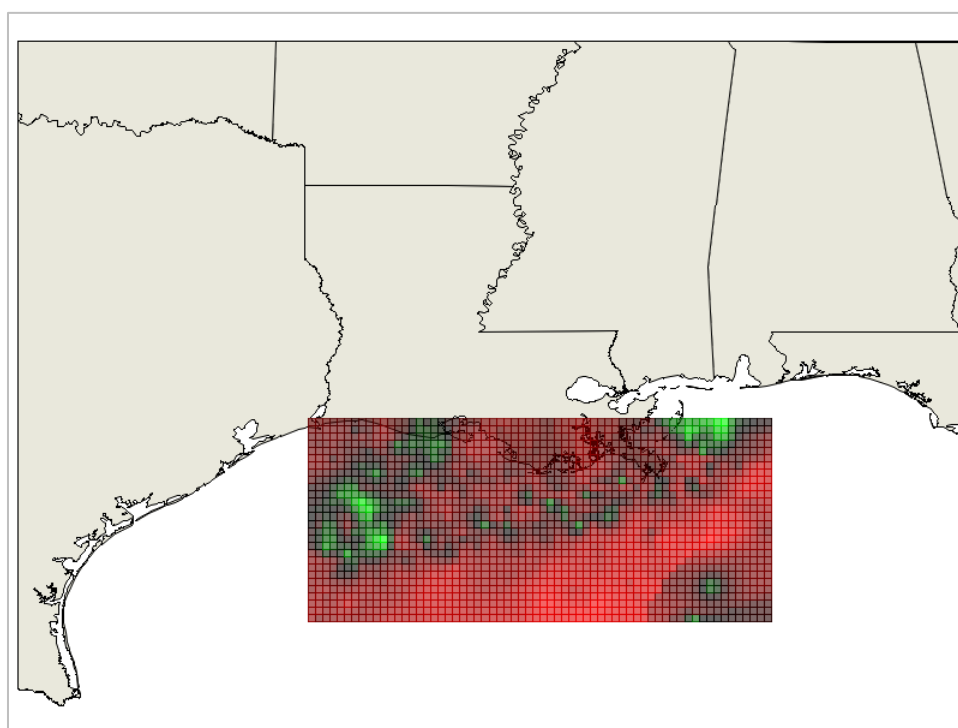
Displaying the map and the annotate grid is simple with the following macro call. It will create an HTML file and an associated PNG image file. It will have a tooltip over the grid area.

```
%draw_map(mystates,mystates,annogrid,state,state,oil,png,"N");
```

The parameters are the map data set, the response data set, the annotate data set, the ID variable, the choro variable, the filename for the HTML file, the device type, and the specification of whether the map is colored ("Y", "N").

All of the macros are found in `mpmacros.sas` that is included with the example code. Appendix 1 shows the macro code.

Figure 3 shows a map of the northern Gulf of Mexico with the results of the oil yield spatial analysis (`logbpm`). Note that the analytical output appears to extend over the land area. Clearly, interpolated values that are farther away from the correlational range of observations are far less dependable and can be ignored and masked out.



**Figure 3. SAS/GRAPH with Annotate Showing logbpm**

### Other Mapping Technology

SAS Bridge for ESRI is an integration between SAS and the Esri's ArcGIS® Desktop product. A similar heat map grid can be created there. To create a heat map grid, perform the following steps:

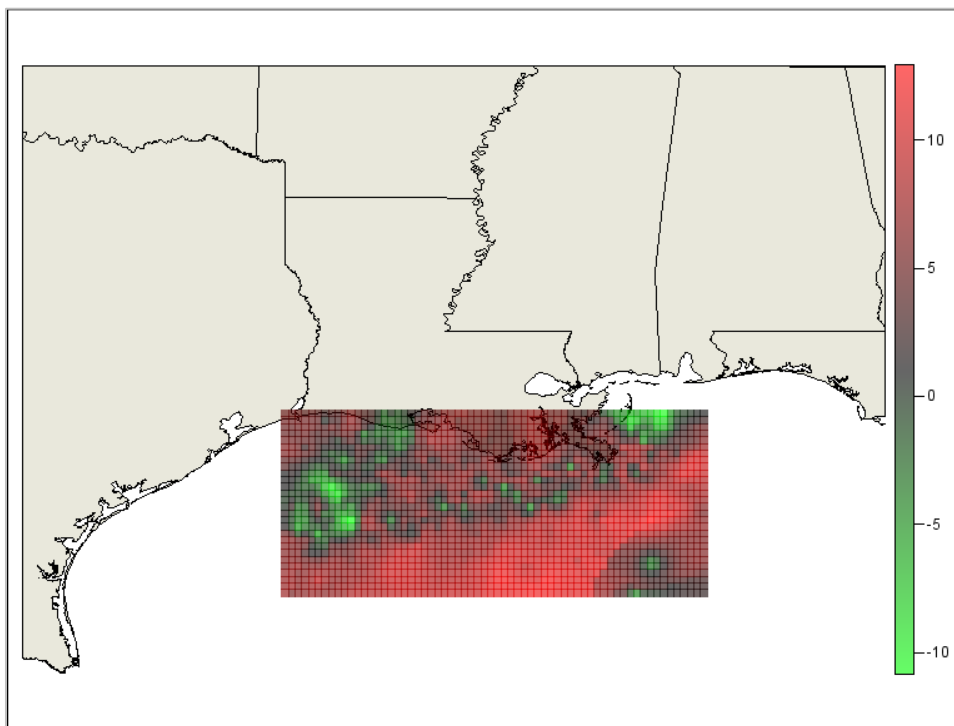
1. Create a polygonal map with the grid location data for use by ArcGIS. This is similar to what was done to create the annotate grid.
2. Assign a unique identifier for each grid cell.
3. Assign the same identifier as the cell to the SAS data set, `bpmPlotData`, in order to join it to the map.

Together at Last: Spatial Analysis and SAS® Mapping, continued

4. From within ArcGIS, open and join the grid map data and the SAS data.
5. Set the color theme values.

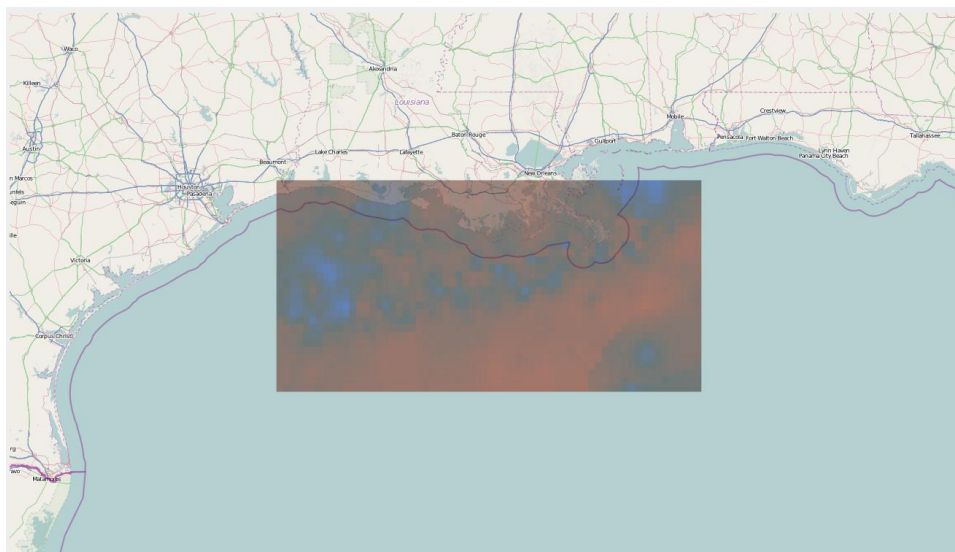
### Future Mapping

Future releases of SAS will have additional methods for creating similar maps. Figure 4 is an example created with ODS Graphics using a map and a heat map. The heat map is used in place of the annotated grid. The map data must be subsetting and prepared as it was in the SAS/GRAPH example.



**Figure 4. ODS Graphics with Map and Heat Map**

Figure 5 is an example of putting a heat map on top of a base image from OpenStreetMap.



**Figure 5. Heat Map on an OpenStreetMap map**



## CONCLUSION

SAS Spatial Analytics and SAS Mapping complement each other. Viewing your results in geographic context adds a new level of understanding to your results. It is simple to view your results on a map as illustrated in this paper. Currently, you can also use SAS Bridge for ESRI to view your results. In future releases, additional mapping technologies will be available.

## REFERENCES

Kolovos, Alexander. 2010. "Everything in Its Place: Efficient Geostatistical Analysis with SAS/STAT Spatial Procedures." *Proceedings of the SAS Global Forum 2010 Conference*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/resources/papers/proceedings10/337-2010.pdf>.

## RESOURCES

Paper and example code downloads.

"Together at Last: Spatial Analytics and SAS® Mapping." SAS Presentations at SAS Global Forum 2012, Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/rnd/papers>

"SAS Mapping: Technologies, Techniques, Tips and Tricks." Papers from SUGI 28, Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/rnd/papers>

"Tips and Tricks II: Getting the most from your SAS/GRAPH maps." Papers from SUGI 29, Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/rnd/papers>

"Tips and Tricks III: More Unique SAS/GRAPH Maps." Papers from SUGI 30, Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/rnd/papers>

"Tips and Tricks IV: More SAS/GRAPH Map Secrets." SAS Presentations at SAS Global Forum 2009. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/rnd/papers>

## CONTACT INFORMATION

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

A. Darrell Massengill  
SAS Institute Inc.  
100 SAS Campus Drive  
Cary, NC, 27513  
919-677-8000  
[Darrell.Massengill@sas.com](mailto:Darrell.Massengill@sas.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.

**APPENDIX 1 - MACRO PROGRAMS****Make\_rectangle**

```

/*-----*/
/* Macro to create rectangle from a x,y locations in
 * the input dataset
 * annodata = annotate data set created.
 * inputdata = Input data set containing points to be drawn.
 *     Note that this data must contain variables x and y
 *     with the location of the points.
 * rwidth = width of rectangle (optional - default value)
 * rdepth = depth of rectangle (optional - default value)
 * lcolor = Line color
 * fcolor = Fill color
 * solid = "Y" if you want a solid/filled rectangle. "N" = empty.
 * center = "Y" if you want rectangles centered at x,y. "N" means
 *     x,y specifies the upper left corner.
 * tipvar = Variable name to be used for tool tip (optional).
 */
%macro make_rectangle(annodata,inputdata,
                     rwidth,rdepth,lcolor,fcolor,solid,center,tipvar);
data &annodata;
  length function $8 color $ 12 style $20 ;
  retain xsys ysys '2' hsys '3' when 'a' size .7;
  retain width 0.01 depth 0.01 ;
  %if (&inputdata ^= ) %then %do;
    set &inputdata;
  %end;
  %if (&solid = 'Y') %then %do;
    style='solid';
  %end;
  %else %do;
    style='empty';
  %end;
  line=1;
  /* optional arguments */
  %if (&rwidth ^= ) %then %do;
    width=&rwidth;
  %end;
  %if (&rdepth ^= ) %then %do;
    depth=&rdepth;
  %end;

  /* optional tool tips */
  %if (&tipvar ^= ) %then %do;
    html= 'title=' || quote("&tipvar" || '=' || trim(left(&tipvar)) );
  %end;
  startx =x;
  starty= y;

  /** create the rectangle ***/
  /* coordinates of the rectangle */
  function='POLY'; color=&fcolor;
  x=startx; y=starty; output; /* start position */
  /*output each corner */
  function='POLYCONT'; color=&lcolor;
  x=startx+width; y=starty; output;
  x=startx+width; y=starty-depth; output;
  x=startx; y=starty-depth; output;
  x=startx; y=starty; output;

run;
%mend;
/*-----*/

```

**To\_degrees**

```

/*-----*/
/* Macro to convert degrees radians to degrees.
 * outdata = output data that is created.
 * inputdata = Input data set containing points to be drawn.
 *     Note that this data must contain variables x and y
 *     with the location of the points.
 */
%macro to_degrees(outdata, inputdata);
/* Convert from radians to degrees */
data &outdata;
    set &inputdata;
    x=-x*45/atan(1);
    y= y*45/atan(1);
run;
%mend;
/*-----*/

```

**Get\_minmax**

```

/*-----*/
/* Macro to find the highest and lowest numbers of a variable
 * inputdata = Input data set containing points to be drawn.
 * varname = Variable to process
 *     Note that this data must contain variables x and y
 *     with the location of the points.
 * NOTE: This macro creates &minval and &maxval with results
 */
%macro get_minmax(inds,varname);

%global minval;
%global maxval;
data tempmm;
    set &inds END=lastob;
    retain maxval -999999999;
    retain minval 999999999;
    if &varname <= minval then
        minval=&varname;
    else if &varname >= maxval then
        maxval=&varname;

    if (lastob = 1) then do;
        output;
        call symput("minval",minval);
        call symput("maxval",maxval);
    end;
run;
%mend;
/*-----*/

```

**Draw\_map**

```

/*-----*/
/* Macro to display a map.
 * mapdata = map data set.
 * respdata = response data set.
 * annodata = annotate data set.
 * id = id statement variable.
 * choro = choro statement variable.
 * name = file name to output.
 * dev = device to draw the map.
 * mapcolor = "Y" to color the map background or "N" to not color
 *          map background (optional - default is "Y").
 * htmlvar = Variable containing html info. (optional)
 */
%macro draw_map(mapdata,respdata, annodata, id, choro, name, dev, mapcolor, htmlvar);
  /*****
  /* Create an HTML file showing the map and points.
  *****/
  GOPTIONS reset=all;
  GOPTIONS CBACK=white;
  GOPTIONS DEVICE=&dev;
  ODS LISTING CLOSE;
  ODS HTML path=odsout body="&name._&dev..html"
  ;
  pattern;
  %if ((&mapcolor ne ) or (&mapcolor = "N")) %then %do;
    pattern v=me c=CX000000 r=100;
  %end;
  %else %do;
    pattern v=s c=CXE9E8DC r=100;
  %end;
  %if (&htmlvar ^= ) %then %do;
    %let htmlv=html=&htmlvar;
  %end;
  %else %do;
    %let htmlv=;
  %end;
  goptions border;

  proc gmap map=&mapdata anno=&annodata data=&respdata all;
    id &id;
    choro &choro /nolegend name="&name" &htmlv;
  run;
  quit;
  ODS HTML CLOSE;
  ODS LISTING;
%mend;
/*-----*/

```