Paper 076-2008

Mapping without GMAP:
An Innovative Use of the SAS/Graph® Annotate Facility

Pete Lund, Looking Glass Analytics, Olympia, WA

ABSTRACT

When you hear “annotate data set” you probably think of putting a reference line on a graph of adding some explanatory text to a plot. These are the most common uses of annotate and the SAS/Graph Annotate facility allows you enhance your graphic output in a variety of ways.

This paper takes Annotate one step further. Rather than using SAS/Graph and PROC GMAP to create a choropleth map and an Annotate data set to enhance it, we’ll do it all with the Annotate facility. An existing map image file (e.g., gif or png) is loaded, using Annotate, and pie and bar charts, also created with Annotate, are added to it. This technique allows for mapping information specific to a particular point on a map, rather than shading or highlighting an area of the map. Also, a “familiar” map can be used – one containing roads, labels, legends, and other features that are difficult to reproduce using PROC GMAP. The techniques described in this paper are extensible to any number of applications and highlight a number of features of the Annotate facility and the SAS-supplied Annotate macros.

A QUICK GUIDE TO THE SAS/GRAPH ANNOTATE FACILITY

SAS/Graph procedures can create many different types of charts, plots and maps. There are a number of mechanisms for adding information to that output including axis, symbol, pattern and legend statement and procedure-specific options. In addition, the SAS/Graph Annotate facility allows you to define commands to create graphics or to “annotate” other SAS/Graph-generated output with additional graphical elements.

Annotate Data Sets

The Annotate commands are stored in SAS data sets. The data set variables have specific names – not all of the variables will be used for all Annotate commands. Some of the variables include:

- Function – the type of Annotate command
- X and Y – specify the x and y coordinates of the output
- Color – the color of the output
- Text – text to display
- Style – font, bar pattern or image type (depending on the command)

If a variable is not used by a particular command it is ignored, as are any non-Annotate variables in the dataset.

Each observation contains information for a single command, specified in the Function variable. It’s often helpful to keep a little history in mind when creating an Annotate data set. Think of the output being generated on a plotter. We might need to move the pen to a specific location on the paper, then draw a line, move the pen again, then add some text, and so on. Commands include,

- Move – moves the “pen” to the specified x,y location
- Label – places text on the page
- Draw – draws a line from the current location to the specified x,y
- Bar – treats the current x,y location as one corner of a bar and the specified x,y as the other corner
- Image – places an image (gif, jpg, png, etc.) on the page

Note: there are a number of Annotate data set variables which define the “environment: of the annotation: the coordinate system to use, whether an element should be placed before or after other graphics, etc. See the Resource section at the end of the paper for details on these variables.

Annotate Macros

There are many different Annotate command and it can be challenging to remember which commands need which variables. Also, a great number of graphic elements require more than one Annotate command. For example, to draw a bar on the page requires a “move” command and a “bar” command. SAS has supplied a number of macros that simplify the process of creating the Annotate dataset.

By default, the Annotate macros are not available to be used. Issue a call to the %Annomac macro to make the library of Annotate macros accessible.

Each macro has the parameters necessary to create the needed command(s). For example, the %BAR macro has the following seven parameters:

- X1,Y1 – the first corner of the bar
- X2,Y2 – the second corner of the bar (diagonally opposite from the first
- Color – the color of the bar
- Line – which lines should be drawn
- Style – the fill style of the bar

A call to this macro generates two observations in the data set – first, a “move” command using the X1,Y1 parameters and then a “bar” command using the X2,Y2 and the rest of the parameters.

Using Annotate Data Sets with SAS/Graph Procedures

As noted above, Annotate data sets are often used with SAS/Graph procedures that generate output. The ANNO= option on the procedure statement references the dataset to be used. Annotate data sets can be displayed on their own with the GSLIDE procedure. GSLIDE creates no output of its own, but will display annotations, as well as titles and footnotes.

This paper is not intended to be a tutorial on Annotate data sets or the Annotate variables, functions and macros, but rather a discussion of a particular use of the Annotate facility. See the resource section at the end of the paper to get more general annotate information.

THE WEBTB PROJECT

The WebTB project, funded by the CDC, is a web-based application used to monitor and report on tuberculosis cases in the US South Pacific Island territories. These include American Samoa, the Commonwealth of the Northern Marinas Islands, the Federated States of Micronesia, Guam, the Republic of Palau and the Republic of the Marshall Islands. Additionally, some of these territories have finer levels of geography specified. For example, data for Micronesia is reported for Chuuk, Kosrae, Pohnpei and Yap Islands.

Currently, paper forms of reported TB cases are faxed to the CDC for data entry. The WebTB application allows users to enter data into a web-based form rather than paper forms and the data can gathered and transmitted electronically. Case information can also be updated and modified as needed.

The application also offers a suite of reporting tools, including demographic and case-related reports. In addition, there is a set of reports that display case-related data on a map. The user can select a time frame, geography and analysis variable on which to report – a table of information and a map are included in the report. The example in Appendix A shows a map detailing the method of case verification for Micronesia.

An example of the parameter selection screen for this map-based report is shown here. Users of the system are able to produce reports for selected time periods, geographies, demographics, and TB-related variables

THE WEBTB MAPS

The map reports contain a table of information, created with PROC REPORT, and a map. The map contains all the information presented in the table, along with a legend.

All of the components of the map were created using Annotate data sets. The information was displayed using PROC GSLIDE.

Placing the Map on the Page

A static map file, in gif format, was obtained for each of the territories listed above. The map of the Federated States of Micronesia is shown here. We used maps that showed common features such as place names, oceans, scales, etc.

Philippine Sea
Northern Mariana Islands (U.S.)
0 350 700 km
0 350 700 mi
North Pacific Ocean
Guam (U.S.)
Yap Islands
Caroline Islands
Pohnpei
Truk Islands (Chuuk)
Kosrae
Equator
South Pacific Ocean
INDONESIA
PAPUA NEW GUINEA
SOLOMON ISLANDS

Map of Federated States of Micronesia

The observations in the Annotate data set for placing the map on the report page are the only ones for which Annotate macros were not used. It takes two Annotate observations to place the map:

- MOVE sets the x,y coordinates of the lower left corner of the graphic

```
x=10; y=0; function='move'; output;
```

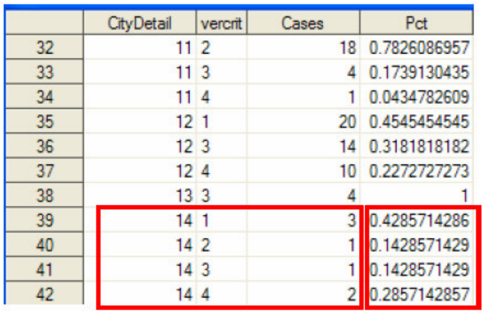
- IMAGE sets the x,y coordinates of the upper right corner of the graphic and names the image file to be displayed.

```
x=60; y=45; function='image';
imgpath='…\fsm.gif'; style='fit'; output;
```

The Analysis Data Set

Notice in the example report shown in Appendix 1 that little donut charts and bar charts will be added to the map. These are created “on-the-fly” based on the geography, time period and breakout variable selected from the web site.

For example, suppose a report is requested for a map of TB cases, broken out by verification method, for the Federated States of Micronesia (FSM). FSM has four regions for which data is collected (see list above). When the request is submitted, the underlying data is summarized to one observation per region (CityDetail) per verification method.(VerCrit). The data set snippet shown here highlights the data for Yap Islands (CityDetail: 14).



	CityDetail	vercrit	Cases	Pct
32	11	2	18	0.7826086957
33	11	3	4	0.1739130435
34	11	4	1	0.0434782609
35	12	1	20	0.4545454545
36	12	3	14	0.3181818182
37	12	4	10	0.2272727273
38	13	3	4	1
39	14	1	3	0.4285714286
40	14	2	1	0.1428571429
41	14	3	1	0.1428571429
42	14	4	2	0.2857142857

There are three things to keep in mind when looking at the Yap Islands data:

1. there are observations for four verification methods (1-4)
2. there were a total of 7 cases
3. the percentages add up to 1

We are going to use this data to create the graphics displayed on the map.

Creating a Pie Chart – the First Slice

The pie chart is created using the %SLICE macro. The macro accepts a number of parameters which describe a pie chart slice:

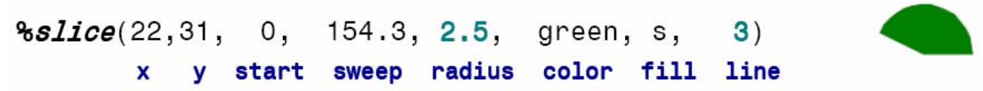
- x,y coordinates of the center of the pie

- starting position of the arc, in degrees (0 is at 3:00)
- sweep of the arc, in degrees
- radius of the circle
- other parameters, including color, fill and line style

Let's take a look at the data for the first Yap Islands observation and the parameters for the macro:
The center of the pie chart for Yap Islands was determined by trial and error, until a suitable position on the map graphic was found.

Since this is the first observation for Yap Islands, a starting position of 0 is used.

The sweep of each pie slice is based on the percentage of cases for the observation. In this case, 42.857% of the cases were for verification method 1 (obs 39 in the data sample above). That equates to a sweep of 154.3 ° (360 ° * .42857).



```
%slice(22,31,  0,  154.3, 2.5,  green, s,   3)
        x   y  start  sweep  radius  color  fill  line
```

The color green is used for the slice and a solid fill pattern is used. Finally, line type 3 designates that a complete border is drawn around the slice.

An observation by the call to the macro with the values passed above.

Creating a Pie Chart – All the Slices

Obviously, we do not want to hard-code the pie slice values, since these maps are generated from the website and we don't know what maps might be requested. Also, we have the data in the dataset to create them – so, let's write some code to create all the slices.

There is one thing that makes this code a bit longer than it would otherwise be. The color parameter of the %SLICE macro must be a constant – all the other parameters may contain variables. This means we need to have a macro call for each color of slice that we might want. With that in mind, here's the code:

```
if first.CityDetail then                              ❶
  do;
    StartPos = 0;
    Coordinates = put(CityDetail,PieCenter.);
    x = input(scan(Coordinates,1,'-'),3.);
    y = input(scan(Coordinates,2,'-'),3.);
  end;

Sweep = 360 * Pct;                                    ❷
Color = put(VerCrit,$PieColor.);                      ❸

if Color eq 'green' then do;                          ❹
  %slice(x,y,StartPos,Sweep,2.5,green,s,3); end;
if Color eq 'red' then do;
  %slice(x,y,StartPos,Sweep,2.5,red,s,3); end;
if Color eq 'yellow' then do;
  %slice(x,y,StartPos,Sweep,2.5,yellow,s,3); end;
if Color eq 'blue' then do;
  %slice(x,y,StartPos,Sweep,2.5,blue,s,3); end;

StartPos = StartPos + Sweep;                          ❺
```

❶ On the first observation for a new geography, we want to reset the start position of the pie to 0 (3:00) We also get the x and y values for the center of the pie from a informat (PieCenter) that maps the CityDetail values to the coordinate values.

❷ The first slice will be 360° times the percentage value in the first observation. In our sample dataset, that would be .42875. The sweep would be 154.3 °.

❸ Then, we need to get the color value for the type of slice we're drawing. In this case, it will be based on the verification method which has a value of "1" in the first observation. We have a format ($PieColor) that maps the value "1" to the value "green" – this will be the color of the first slice.

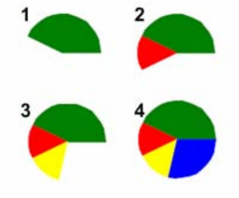
❹ Now that we have the x, y, sweep and color values it's time to draw the bar. The only difference in these four calls to the %SLICE macro is the color value.

❺ After the %SLICE macro writes the observation, we need to set the start value of the next slice. It will be the current start value (0 on the first slice) plus the sweep of the current slice (154.3 on the first slice). So, in our example the second slice will have a starting position of 154.3 and the sweep will begin at that point.

How Does the Pie Look?

When we run the code above for the Yap Islands observations, the calls to the %SLICE macro would be as follows. Notice that the only differences are the starting position, the sweep and the color – the x, y, fill and line type are the same for each slice.

```
%slice(22,31,    0,154.3,2.5,green,s,3)     1
%slice(22,31,154.3, 51.4,2.5,red,s,3)       2
%slice(22,31,205.7, 51.4,2.5,yellow,s,3)    3
%slice(22,31,257.1,102.9,2.5,blue,s,3)      4
```



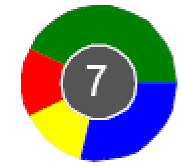
Since the percentages for each location sum to 1, we always get a complete pie, even if there are fewer than four slices.

Turning the Pie into a Donut

Notice in the Appendix A sample report that the pie charts are really donut charts, with a hole in the center that contains the total number of cases. We actually will use the %SLICE macro again to create the donut center.

We only need to create the center once for each geography and can use the same x and y values as we did for the slices. We accumulate the total number of cases as we process the data set and then run the following code:

```
TotalCases + Cases;                                        ❶

if last.CityDetail then                                    ❷
  do;
    %slice(x,y,0,360,1.2,gray55,s);                        ❸
    %slice(x,y,0,360,1.2,white);                           ❹
    %label(x,y,TotalCases,white,0,0,1,'Arial',5);          ❺
  end;
```



❶ Accumulate the total number of cases for the current geography.

❷ The donut center code only needs to be run once for each geography, after the pie slice observations have been written and the total number of cases accumulated.

❸ First, draw a 360° solid gray slice – in other words, a solid gray circle. It has the same x, y center as the pie slices and the radius is just under half what the pie slice radius was (1.2 vs. 2.5).

❹ Next, draw another 360 ° "slice" – this one is white and not filled. What this gives us is a white border around the gray circle, which helps the center to stand out a bit more.

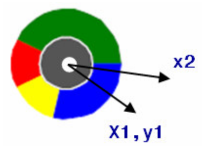
❺ Finally, the %LABEL macro is used to put the total case count in the center of the circle, in white Arial text.

Creating a Bar Chart

We want to have a little more information on the map. We'll use a bar chart to indicate the rate of TB cases per 100,000 population. The bar heights on the map are relative to each other. The largest rate will have a bar that's 80% as high as the pie chart. All the other bars are proportional in size to that highest rate. There is preliminary process that computes the highest rate for the report and stores it in the macro variable &MaxRate.

The placement of the bar chart is tied to the same x, y as the center of the pie chart. We'll use the %BAR macro to draw the bar. The position of the bar is defined by defining the x, y coordinates of the lower left corner and upper right corner of the bar. Let's give some thought to those values. The width of all the bars will be the same, so we can have a "fixed" set of x values. Also, the lower y value is going to be the bottom of the bar, so its value is also data independent.

```
x1 = x + 3.5;
x2 = x + 5.5;
y1 = y - 2.5;
```



The x and y values in the equations to the left are the coordinates of the pie center.

Remember that the radius of the pie was 2.5 – here we set the bars inner x value (x1) to x + 3.5 – this will place the bar just to the right of the pie. We want the bar to have a width of 2, so the outer x value (x2) is set to x + 5.5. Also, we want the bottom of the bar to be at the same level as the bottom of the pie. Again, the pie has a radius of 2.5, so the bottom y of the bar (y2) has a value of y – 2.5. We now have three of the four coordinates for the bar.

The top y value is based on the maximum rate value for the geographies to be mapped.

```
y2 = round(4 * (Rate / &MaxRate),.01) + (y - 2.5);
```

The pie has a diameter of 5 units. The first part formula above will give the highest rate a value of 4 units, or 80% of the pie diameter, since Rate and &MaxRate would have the same value. This is added to the bottom y value (y – 2.5).

Now we have all four coordinates needed to define the corners of the bar. The %BAR macro will write two observations to the annotate data set: a MOVE function observation, which will "move the pen" to the lower left corner (x1,y1) of the bar and a BAR function observation which will draw a bar to the upper right corner (x2,y2).

```
%bar(x1,y1,x2,y2,purple,0,s);
```

The last two parameters in the %BAR macro are the line type and fill style; "0" denotes that a border should be drawn around the entire bar and "s" denotes a solid bar should be drawn.

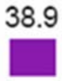
Finally, we'll use one more annotate macro to label the bar. The %LABEL macro will write text at a given x, y position. In this case, we want the text centered above the bar, so our x is between the values set above for x1 and x2:

```
x3 = x + 4.5;
```

The y position for the top of the bar (y2) can be used for the label. One of the %LABEL macro parameters is the position of the text relative to the x, y coordinate: a value of "2" denotes that the text should be centered above the point – that will place our text above the bar.

The text is stored in a data set variable called RateText and the following macro call creates the label observation.

```
%label(x3,y2,RateText,black,0,0,1,'Arial',2);
```



The parameters between the text color and font specification are the angle, rotation and size of the text.

The Entire Data Step

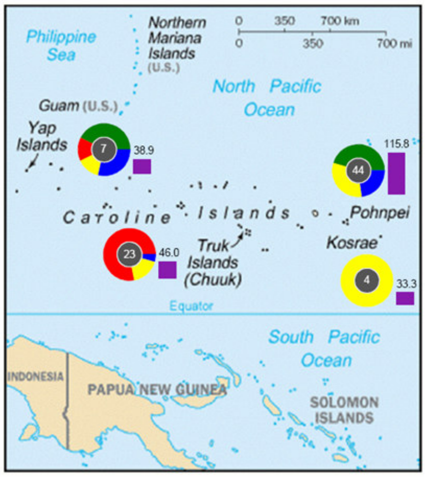
As noted earlier, the data step logic is broken down into those tasks that need to be done once and those that are done for each observation of the data set. Here is a shell of the entire data step, with placeholders for the logic that has just been discussed.

```
data MapAnno;
  set AnalysisBreakdown;
  by CityDetail;

  if _n_ eq 1 then
    do;
     <put map on page>
    end;

  <build pie slices>

  if last.CityDetail then
    do;
      <add donut center>
      <build bar chart>
    end;
run;
```

The map is placed on the page at the beginning of the data step

Each observation in the dataset contains the information for one pie slice

When all the slices for a region have been defined, the donut center and bar chart observations are written to the annotate dataset



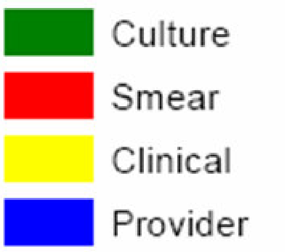
Creating a Legend

We'll use similar techniques to create another Annotate data set containing commands to build a legend for the map. Just like the pie and bar charts, there are some observations in this data set that are data-driven, In this case, the breakout variable selected will determine the number and labeling of the legend entries.

Notice the table at the top of the report in Appendix A. The verification method columns are labeled with a format. We'll use PROC FORMAT to create a data set containing those labels to build our legend. Also, we'll take advantage of certain aspects of the legend to simplify the code: first, the legend will always be in the same place on the page, so x and y coordinates can be fixed. Second, the color sequence (green, red, yellow, blue,…) will always be the same, no matter what breakout variable is selected.

In our example, the format dataset (AV_Format) will have four observations, one for each verification method. The variable called “label” contains the text used in the table and now in the legend. The following code generates the legend bars:

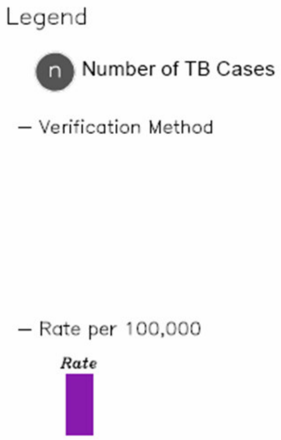
```
if _n_ eq 1 then do;
   %bar(6,66,15,71,green,1,s);
   %label(17,69,Label,black,0,0,4,'Arial',6);
end;
if _n_ eq 2 then do;
   %bar(6,59,15,64,red,1,s);
   %label(17,62,Label,black,0,0,4,'Arial',6);
end;
if _n_ eq 3 then do;
   %bar(6,52,15,57,yellow,1,s);
   %label(17,55,Label,black,0,0,4,'Arial',6);
end;
if _n_ eq 4 then do;
   %bar(6,45,15,50,blue,1,s);
   %label(17,48,Label,black,0,0,4,'Arial',6);
end;
```



The format data set has been ordered to match the sequence of slices, so in the first iteration of the data step (_n_ = 1), the green bar is drawn. This pattern continues for all the observations in the data set. So, we’ll only get as many bars drawn in the legend as we have values for the breakout variable. Notice that everything is hard-coded except for the text parameter of the %LEGEND macro calls. This is the value that comes from the data set.

In addition to the bars, there are a number of “static” elements to the legend – things that are the same across all the reports. These include the section headings, a sample donut and a sample bar. The commands to create all of these pieces is done once, after the label data set is processed.

```
if done then
 do;
   %label(1,98,'Legend',black,0,0,5,'Arial/bold',6);

   %slice(10,86,0,360,5,gray55,s);
   %label(10,87.5,'n',white,0,0,5,'Arial/bold',5);
   %label(15,87.3,'Number of TB Cases',black,0,0,4,'Arial',6);

   %label(3,76,"- &AV_Name",black,0,0,4,'Arial',6);

   %label(3,y1,'- Rate per 100,000',black,0,0,4,'Arial',6);

   %label(14.5,y2,'Rate',black,0,0,3,italic,2);
   %bar(12,y3,17,y2,Purple,1,s);
 end;
```



Again, notice that almost everything here is hard-coded. The only exception is the header for the bars. The macro variable &AV_Name contains a label for the breakout variable.

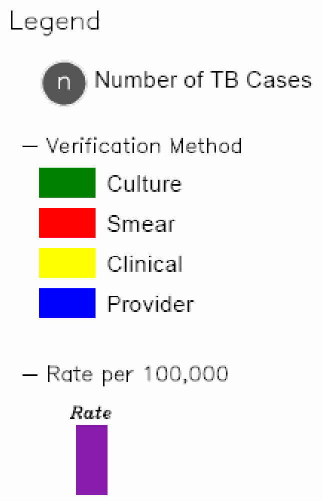
The Legend Data Step

The legend data set is built like the pie/bar data set was, with both static and dynamic pieces. Like shown above, here is a shell of the legend data step.

```
data LegendAnno;
  set AV_Format end=done;

  <build value-specific bars>;

  if done then
    do;
      <legend label>;
      <section labels>;
      <donut center>;
      <bar chart>;
    end;
run;
```

The legend bar labels are created from the dataset, which has an observation for each bar

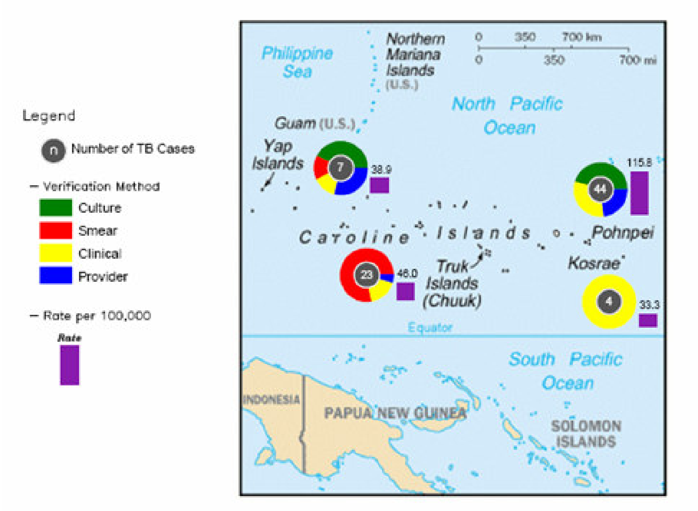
All the other pieces of the legend, the section headers, sample donut and sample bar are all done once, at the end of the data step processing



Putting It All Together

The two annotate datasets are placed on the graphics page with separate PROC GSLIDE procedures, each in its own graphics region.

```
goptions vorigin=1.7in vsize=3.5in
         horigin=0.5in hsize=2.7in;

proc gslide anno=LegendAnno;
run;  quit;


goptions vorigin=1.0in vsize=6.0in
         horigin=2.0in hsize=6.0in;

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



Conclusion

There's a lot more to the SAS/Graph Annotate facility than meets the eye. The ability to create your own graphics and import you own images opens up a myriad of possibilities for presenting information. Also, the Annotate macros make it easy to create the data sets you need – let me know where your own creativity leads you.

RESOURCES

There are a couple of resources that will help you in your quest to learn about the SAS/Graph Annotate facility.

Annotate: Simply the Basics, by Art Carpenter, SAS Publishing, 1999

The SAS Online Docs as http://support.sas.com/onlinedoc/913/docMainpage.jsp

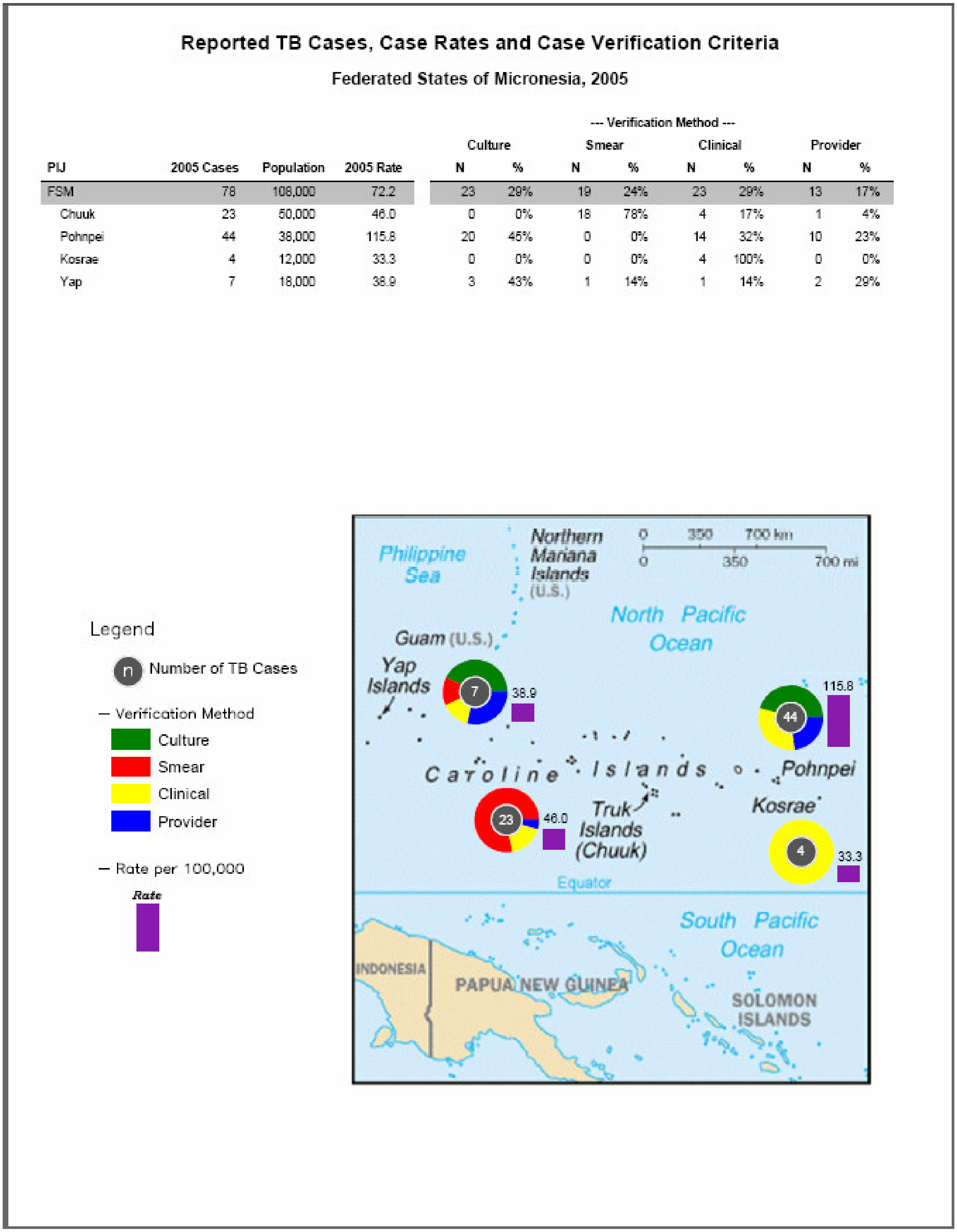
- Click on SAS/Graph Reference and then, The Annotate Facility

AUTHOR CONTACT INFORMATION

Please let me know other things you've thought of to do with the Annotate Facility!

Pete Lund
Looking Glass Analytics
215 Legion Way SW
Olympia, WA 98501
360-528-8970 voice
360-570-7533 fax
pete.lund@lgan.com
www.lgan.com

APPENDIX A – SAMPLE OF WEBTB REPORT



Reported TB Cases, Case Rates and Case Verification Criteria

Federated States of Micronesia, 2005

				--- Verification Method ---							
				Culture		Smear		Clinical		Provider	
PIJ	2005 Cases	Population	2005 Rate	N	%	N	%	N	%	N	%
FSM	78	108,000	72.2	23	29%	19	24%	23	29%	13	17%
Chuuk	23	50,000	46.0	0	0%	18	78%	4	17%	1	4%
Pohnpei	44	38,000	115.8	20	45%	0	0%	14	32%	10	23%
Kosrae	4	12,000	33.3	0	0%	0	0%	4	100%	0	0%
Yap	7	18,000	38.9	3	43%	1	14%	1	14%	2	29%

Legend

- ● n Number of TB Cases
- Verification Method
  - 🟩 Culture
  - 🟥 Smear
  - 🟨 Clinical
  - 🟦 Provider
- Rate per 100,000
  - *Rate* 🟪