PICTURING PRODUCTION DATA USING BASE SAS® AND SAS/GRAPH® FEATURES

Tamara L. Siler, Exxon Company, U.S.A.

Abstract
As part of regular reviews, we must be able to present large amounts of production data in an informative, timely fashion. To assist us in this we have developed an application using data management and graphics techniques offered by Base SAS and SAS/Graph. Several procedures will be discussed, including PROC FORMAT, PROC GPLOT, and PROC GREPLAY. In using these features, we have truly made our "picture worth more than a thousand words".

Introduction
In the Alaska Interest Organization, we co-manage several fields on the North Slope of Alaska. One field in particular, Endicott, yields not only monthly data, but daily values as well. During reviews, the data needs to be represented in a way such that the optimum number of variables are captured most effectively. Since these reviews often take place with little forewarning we have developed an application in SAS which allows us to manipulate the data while presenting a multipurpose and visually appealing picture.

Our application is composed of several data manipulation and plotting routines that write to a graphics file for later use with PROC GREPLAY. Since the programs are relatively similar, this paper will touch on the special features used without looking at each program individually.

The Procedure

Data input and Variable Creation

data a;
  infile in;
  input date mmddyy8. oilrate gasrate gasflare fue/gas wtrprod count ng/ flood;
go=(gasrate/oilrate)*1000000;
watercut=wtrprod/(wtrprod+oilrate)*100;
cumoil=oilrate;

As part of reading in the data a date format has been assigned to the appropriate parameters (month, day, year). Several base variables are used in equations to set new, more applicable variables (go, watercut). The sum statements allows the calculation of a cumulative variable, cumoil.

Formatting the Date Variable

proc format;
  value AAAA '01OCT87D-31OCT87D=O'
  '01NOV87D-30NOV87D=N'
  '01DEC87D-31DEC87D=O'
  '01JAN88D-31JAN88D=1988'
  '01FEB88D-29FEB88D=F'
  '01MAR88D-31MAR88D=M'
  '01APR88D-30APR88D=A'
  (more formatted dates)
  '01JAN92D-31JAN92D=1992'
  '01FEB92D-29FEB92D=F'
  '01MAR92D-31MAR92D=M'
  '01APR92D-30APR92D=A';

User-defined formats not only allow the transfer of numeric data to character data, but allow user control over the format of variable values which should not be set by SAS defaults. The use of PROC FORMAT shortens the date value so that more dates can be plotted.

1081
A FORMAT statement if used later within the GPLOT statement to refer to the pre-defined format. The statement will read:

```
format date AAAA;
```

Note the period; this must be used when referring to a format outside of a PROC statement.

Placing Text Annotation Directly Onto The Plot

```
data labels;
  length text $ 12.0 color $ 5.0;
  function='/abel';
  position='2';
  xsys='2'; ysys='2'; when='t'; size=.75;
  x='15DEC87D'; y=0; text='WATER (KB/D);'
    style='XSWISS'; color='BLUE';
    output;
  x='21JAN88D'; y=25; text='NO. OF WELLS';
    style='XSWISS'; color='BLACK';
    output;
  x='21JAN88D'; y=45; text='GAS (MSCF/D);'
    style='XSWISS'; color='RED';
    output;
```

The ANNOTATE function allows the placement of text directly on the plot. DATA LABELS defines all the necessary parameters for use with the ANNOTATE function:

1. The LENGTH statement overrides any defaults that might have been inadvertently set.
2. FUNCTION defines the action to be performed.
   FUNCTION='LABEL' places text on the plot area.
3. POSITION controls placement of the text string.
   POSITION='2' means one cell above centered.
4. XSYS and YSYS specify which coordinate system is to be used for X and Y. '2' means the coordinates will be in reference to the data values.
5. WHEN specifies when the function is to be performed in relation to generating other graphics output for the procedure. 'A'==> after the graph is drawn.
6. X and Y are reference point data values.
7. TEXT, COLOR and SIZE refer to the content and appearance of the label.
8. OUTPUT writes the current observation to the data set being created.

As part of the plot statement, the option Annotate='labels' will be used, where 'labels' refers to the name of the dataset created by this procedure.

Sending the Plot to an Output File

```
proc gplot data=a gout:tam.newprod;
```

As part of the PROC GPLOT procedure used to plot desired variable, you can specify a graphics catalog. GOUT=graphicscatalog stores the output produced by GPLOT for later replay. In this case, TAM represents a fully qualified dataset name and NEWPROD is the member that will contain the specified images.

Creating Custom Axis Labels

```
axis1 label=(f=duplex h=1.0 'DATE')
  order='O1OCT87D to O1OCT91D by QTR;
axis2 label=(f=duplex h=1.0 c=green r=90 a=90 
  'OIL KB/D')
  order=0 to 200 by 25;
axis3 label=(f=duplex h=1.0 c=red r=90 a=90 
  'GAS MSCF/D')
  order=0 to 200 by 25;
```

1082
The AXIS statement allows user control over fonts, size, color, length, rotation, and angle of axis titles. It is also helpful in ordering unusual or non-obvious axis divisions, especially dates. Instead of setting up range in HAXIS and VAXIS statements as part of the PLOT statement, a reference is made to the appropriate AXIS statement. In Version 5.18 the AXIS statement must be part of the GPLOT procedure. In Version 6.06, the AXIS statement may stand independent of the GPLOT procedure.

Using PROC GREPLAY

```
x alloc l(tam) da('AIOTLS.SAS.REPLAY3') reuse;
x alloc f(gsasfile) ysysout(a) dso(lm78)
xexec(132) bsize(1320) recfn(b);
goptions device=TCX4510 gaccess=gsasfile
promptchar=0000010000400000X;
proc grep/ay;
```

Once all the desired plots have been generated, PROC GREPLAY can be run. This job allocates the graphics catalog, referred to as GSASFILE. In this case our output device is a color ink-jet plotter, but we also have this procedure with a 36-in color electrostatic plotter as the output device, if larger plots are desired. Once run, this program will activate the PROC GREPLAY panel where the plots can be assigned to windows.

Here is the entry panel for GREPLAY. IGOUT is the input graphics catalog. It is filled in with TAM.NEWPROD, the file each job defined as the destination for output graphics. As previously seen, TAM is the file reference name for 'AIOTLS.SAS.REPLAY3'. Our template catalog (TC) is stored there also, under the member name TEMPLATE. More than one template can be defined in a member; our desired screen divisions are in TEMP1. To view all available templates, use PF2.

```
<table>
<thead>
<tr>
<th>Panel</th>
<th>Clip</th>
<th>Color</th>
<th>L-left</th>
<th>O-left</th>
<th>U-right</th>
<th>L-right</th>
<th>Scale</th>
<th>X-x</th>
<th>X-y</th>
<th>X-x</th>
<th>X-y</th>
<th>X-x</th>
<th>X-y</th>
<th>X-x</th>
<th>X-y</th>
<th>X-x</th>
<th>X-y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WHITE</td>
<td>50.0</td>
<td>50.0</td>
<td>100.0</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>WHITE</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>50.0</td>
<td>T</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>T</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>WHITE</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>50.0</td>
<td>T</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>T</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>T</td>
</tr>
<tr>
<td>4</td>
<td>WHITE</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>50.0</td>
<td>T</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>T</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>WHITE</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>50.0</td>
<td>T</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>T</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>100.0</td>
<td>X</td>
<td>X</td>
<td>T</td>
</tr>
</tbody>
</table>
```

Taking a look at template TEMP1, one can see the screen is divided into sections by using values from 0 to 100 for minimum and maximum screen values.

Panel 1 has these values:
- Lower left corner $x=50.0, y=50.0$
- Upper left corner $x=50.0, y=100.0$
- Upper right corner $x=100, y=100$
- Lower right corner $x=100.0, y=50.0$

1083
These values define Panel 1 as being the upper right quadrant of the screen. This procedure can be used to divide the display area into quadrants, halves, thirds, or other geometric shapes, as long as the correct coordinates can be provided.

After exiting back to the plot selection panel, we can put the plots into windows by referring to the appropriate panel numbers. For this application, the window is divided into quadrants and the plots are placed in Panel 3 (Lower Left Quadrant), Panel 1 (Upper Right Quadrant), Panel 2 (Upper Left Quadrant), and Panel 4 (Lower Right Quadrant). When ‘ENTER’ is pressed, one plot is created. To create more than one plot, just repeat the selection process and hit ‘ENTER’. You may do this as many times as desired. Use PF3 to exit and the plots will then be written to the output device.

Conclusion

This application is easy to maintain and easily adaptable to other data we may receive. PROC GREPLAY continues to be a powerful tool in extending the power of SAS/Graph. The formats and other features mentioned only begin to scratch the surface of the versatility SAS offers.

Acknowledgements

SAS and SAS/GRAPH are registered trademarks of SAS Institute, Inc., Cary, NC, U.S.A. ® indicates USA registration.

The author wishes to thank Ray Missman, Mickie Irvine and Lisa Wood, all of Exxon Company, U.S.A. for their technical support. Thanks also to Hanan Threadgill for her drafting and graphics artist assistance.

Author Contact

For any questions or comments contact:

Tamara L. Siler
Exxon Company U.S.A.
800 Bell, Rm. 3039H
Houston, Texas 77002
(713) 656-7028
Fax No. (713) 656-7100
The Results

Note: For the sake of confidentiality, random values have been chosen to represent the real data.