SAS\GRAPH - 2 Three-dimensional Graphs and Contour Plots

In this section PROC G3D and PROC GCONTOUR are demonstrated to learn the basics of these procedures.

Three-dimensional Graphs
You may choose either a scatterplot of the data or a plot of a three-dimensional surface. The SAS procedure for this is the G3D procedure. The simpler plot is the scatterplot.

The syntax of the procedure is:

PROC G3D DATA=*tablename* ;
SCATTER y * x = z </options> ;

The SCATTER statement allows one to produce a three-dimension scatterplot of data. The "z" variable is plotted on the vertical axis on the page (or screen). Options on the SCATTER statement allow one to select a plotting symbol, plotting color, and more.

One or more options on the SCATTER statement that control the plot appearance are:

COLOR="*color"* Stick to the color basics used so far: RED, BLUE, ORANGE, BLACK, GREEN, GRAY, PURPLE, PINK, YELLOW and BROWN. All points plotted will be the color specified. Quotation marks are required.

COLOR=*color-varible* allows for a character variable in the data table to identify multiple colors on a single scatterplot.

NONEEDLE By default G3D draws a needle from the plotting symbol "down" to the x*y plane. NONEEDLE suppresses this.

SHAPE="*symbol"* identifies the plotting symbol used in the scatterplot. Symbols include: BALLOON, CLUB, CROSS, CUBE, CYLINDER, DIAMOND, FLAG, HEART, PILLAR, POINT, PRISM, PYRAMID, SPADE, SQUARE, and STAR. The NONEEDLE option is available for all symbols except PILLAR and PRISM. All points plotted will be the symbol shape specified. Quotation marks are required.

SHAPE=*symbol-variable* allows for a character variable in the data table to identify multiple symbols on a single scatterplot.

SIZE=*symbol-size* identifies a numeric value which determines the size of all plotting symbols on the graph.

SIZE=*size-variable* allows for a character variable in the data table to identify multiple sizes for plotting symbols on a single scatterplot.

ANNOTATE, ROTATE, TILT are other appearance options that can be specified.

Options that modify one or more of the axes are:

CAXIS=*color* specifies the color of axis lines, tick marks, and grid lines

CTEXT=*color* specifies the color for text on the axes.

GRID draws reference lines from each tick mark on each axis

XTICKNUM=*number* specify the number of major tick marks that are located on a plot's X axis. The default value is 4.

YTICKNUM=*number* specify the number of major tick marks that are located on a plot's Y axis. The default value is 4.

ZTICKNUM=*number* specify the number of major tick marks that are located on a plot's Z axis. The default value is 4.

ZMAX=*value* specifies the maximum data value that is displayed on a plot's Z axis. You can use the ZMAX= option to extend the Z axis beyond the value range. The value that is specified by the ZMAX= option must be greater than that specified by the ZMIN= option. If you specify the ZMAX= option within the range of the Z variable values, the plot's data values are clipped at the level you specified.

The default is the observed maximum value of the z-variable.

There is no similar maximum option for either the X or the Y axes.

ZMIN=*value* specifies the minimum value that is displayed on a plot's Z axis. Defining the ZMIN= value less than the minimum value in the input data set extends the plot's Z axis. Defining the ZMIN= value greater than the minimum value in the input data set displays all Z values in the range of ZMIN-to-ZMAX, and might cause data clipping. .

The default is the observed minimum value of the z-variable.

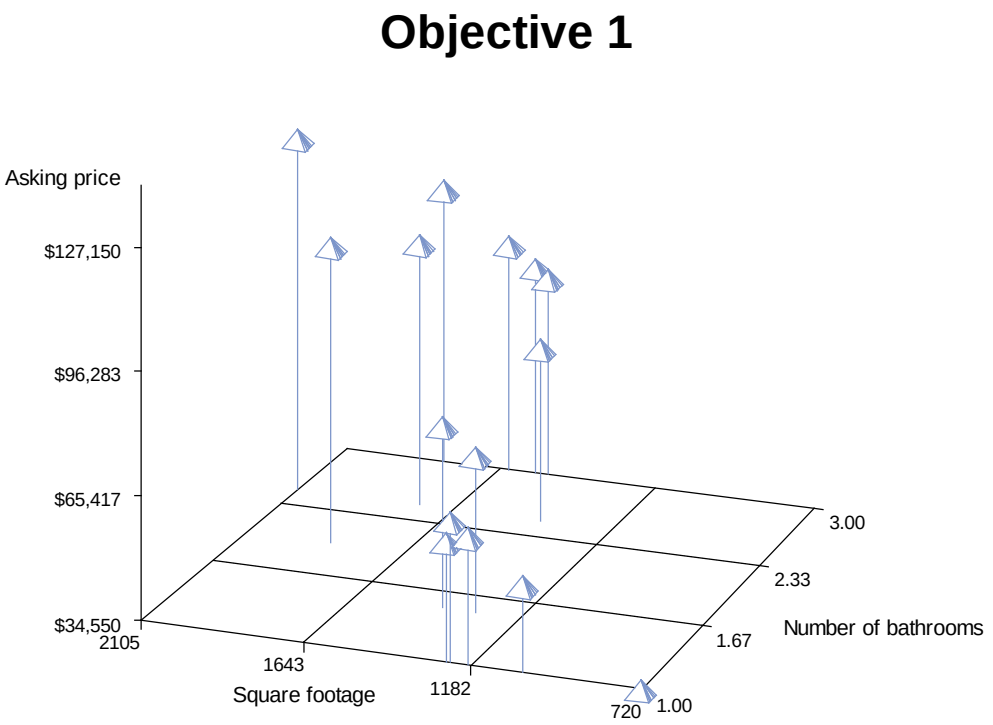
There is no similar maximum option for either the X or the Y axes.

XAXIS= , YAXIS= , ZAXIS= are additional options that can be used in conjunction with global AXISn statements.

The objectives in this section require the HOUSES data set.

Objective 1*: C*reate a three-dimensional scatterplot of the variables price, square footage, and number of baths in the HOUSES data set.

```
PROC G3D DATA=class.houses ;
SCATTER sqfeet * baths = price ;
TITLE 'Objective 1';
RUN;
```



Notice the default settings here for color (pale blue), needles are "on", and the plotting symbol is a pyramid. Also note which axis is the Y axis and which is the X axis.

Note that the axes are labeled with the value of the variable in this demonstration. This is due to the fact that a LABELS statement was used in the creation of the permanent SAS data set HOUSES.

Objective 2: Modify this graph by changing the tick marks on the axis for BATHS to read 1, 1.5, 2 , 2.5, 3. Change the PRICE axis so that the scale begins at 30,000. Change the plotting symbol to a yellow star. Remove the vertical lines or needles from the plotting symbol down to the XY plane. Add gridlines.

```
PROC G3D DATA=class.houses ;
SCATTER  sqfeet * baths = price  /  XTICKNUM=5        ZMIN=30000
                                    SHAPE="STAR"      COLOR="Red"
                                    NONEEDLE          GRID ;
TITLE 'Objective 2';
RUN;
```

You may notice that the needles used in the plots contribute to the depth perception of the plot as does the use of grid lines. For your own application, experiment with these options to produce the image that best conveys the trend in your data. You should also know that when producing a scatterplot for data that contain multiple responses at the same X, Y setting, only one observation is plotted for this setting. This typically is the last observation at this X,Y coordinate setting.

Plot of a three-dimensional surface

The G3D procedure can also be used to plot a three dimensional surface. A PLOT statement is used rather than a SCATTER statement. In order to graph a surface, one needs many x,y coordinates at which z-values are determined. Though not always required, one should run a G3GRID procedure prior to generating a surface plot in G3D. The G3GRID procedure places the data in an order that SAS/G3D can more efficiently use to produce a surface. G3GRID does not generate anY information in an output window, but simply works on the data table that is to be plotted. The GRID statement in G3GRID should match the PLOT statement of G3D.

PROC G3GRID DATA=*tablename*;
GRID y * x = z ;

PROC G3D DATA=*tablename* ;
PLOT y * x = z </options> ;

One or more options on the PLOT statement that control the plot appearance are:

CBOTTOM=color specifies the color of the bottom of the plotted surface.

CTOP=color specifies the color of the top of the plotted surface.

SIDE requests a side wall of the surface.

XYTYPE=0 |1 | 2 | 3 specifies the direction of lines that are used to represent the plot's surface. Both X and Y are displayed by default. The valid values for the XYTYPE= option are as follows:

1. XYTYPE=0 (Java and ActiveX only) No lines are displayed. The plot is displayed as a solid surface.
2. XYTYPE=1 draws lines that are parallel to the X axis. The surface is displayed by using lines that represent Y axis values.

3. XYTYPE=2 draws lines that are parallel to the Y axis. The surface is displayed by using lines that represent X axis values.

4. XYTYPE=3 draws lines that are parallel to both the X and Y axes. Displays the surface by using lines that represent values for both X and Y.

The axis options specified for the SCATTER statement are applicable to the PLOT statement also. (CAXIS= , CTEXT= , GRID, XTICKNUM, YTICKNUM, ZTICKNUM, ZMIN, ZMAX).

Other PLOT statement options include ANNOTATE, TILT, ROTATE, XAXIS, YAXIS, and ZAXIS options.

Objective 3: Create a surface plot for three-dimensional equations.

To demonstrate this, a data set containing many points across and XY grid will first need to be generated.
There will be two surfaces will be plotted.

1. $z = x^3 + 3xy^2 + 3y^2 - 15x$

2. $z = 3x - x^2 - y^2$

In the Enhanced Editor create two temporary SAS data sets, one for each of the two surfaces.

```
DATA  one;
x = -2.5;
y = -1;
  DO x = -2.5 to 2.5 by 0.1;
      DO y = -1 to 1 by 0.1;
            z = x*x*x + 3*x*y*y + 3*y*y - 15*x;
            output;
      END;
  END;

DATA two;
x = -3;
y = -3;
  DO x = -3 to 3 by 0.2;
      DO y = -3 to 3 by 0.2;
            z = 3*x - x*x - y*y;
            output;
      END;
  END;

RUN;
QUIT;
```

In the above two DATA step operations, the starting values for X and Y must be specified. For any surface you wish to plot, you will need to examine your data for logical starting values (and stopping values) for both X and Y. The two DATA step operations demonstrate a DO loop. For every DO statement, you must have and END statement. The general syntax of a DO loop is this:

DATA a;
x = *starting value;*
DO x = *starting value* TO *stopping value* BY *interval;*
<insert transformation>
OUTPUT;
END;

You do not have to specify BY *interval*. If you do not, SAS assumes an interval of 1. In data set ONE above x takes the values -2.5, -2.4, -2.3, . . . , 0, . . . 2.3, 2.4, 2.5.

Caution: If you specify an interval that is too small for the starting and stopping values, the DATA step will take a very long time to run. When you run nested DO loops for two variables as is done above, the number of X,Y pairs used is the product of the number of x values times the number of y values. In data set one above, there are 51 * 21 = 1071 points on the XY plane for which a z value is determined.

```
PROC G3GRID DATA=one;
GRID y * x = z ;

PROC G3D DATA=one;
PLOT y * x = z ;
TITLE  'Objective 3';
RUN ;
```

The size of the "mesh" on the surface corresponds with the interval size chosen in the DO LOOPS.

For the second surface

```
PROC G3GRID DATA=two;
GRID y * x = z ;

PROC G3D DATA=two;
PLOT y * x = z ;
RUN ;
```

Modify these graphs using GRID or SIDE options. What happens if you switch x and y in the G3D procedure alone? Anything? Nothing?

Graphs – Contour Plot
Frequently, contour plots are presented with or instead of surface plots. These are generated by the GCONTOUR procedure.

PROC GCONTOUR DATA=*tablename* ;
PLOT y * x = z </options> ;
RUN;

- contour options:

  CLEVELS=color(s)
  JOIN
  LEGEND=LEGEND<1...99>
  LEVELS=value-list
  LJOIN
  LLEVELS=line-type-list
  NLEVELS=number-of-levels
  NOLEGEND
  PATTERN
  SMOOTH

- labeling options:

  AUTOLABEL

Objective 4: Produce a contour plot in WORK.ONE using the GCONTOUR procedure where the contours are defined in a legend. Note the default settings for the number and style of the contours and the default legend.

```
LEGEND;
PROC GCONTOUR DATA=one;
PLOT y * x = z ;
TITLE 'Objective 4';
RUN;
```

Objective 5: Insert the AUTOLABEL option on the PLOT statement in the GCONTOUR procedure. Submit the program and observe the resulting image. Then insert the NOLEGEND option after the AUTOLABEL option on the PLOT statement.

Objective 6: Note that there were seven contours drawn. You can select the number of contours you'd like in the image. Add NLEVELS = 10 to have SAS determine ten contours for the image.

Objective 7: Change the levels at which the contour lines are drawn to: -30, -25, -20, -15, -10, -5, 0, 5, 10, 15, 20, 25, 30. (The symmetry about zero is used in this example for illustration only. Select contour levels based on the equation you wish to graph.) Delete NLEVELS=10 to

LEVELS = -30 -25 -20 -15 -10 -5 0 5 10 15 20 25 30. Submit the program and observe the change in contours. Note what happens when you select a contour level that goes outside the range of your data.