INSIGHT into Multiple Regression
Robert A. Muenchen, University of Tennessee

Introduction

Two dimensional scatter plots are considered an essential part of building linear models; how better to see if your model fits than to run the predicted line through the data? The SAS/INSIGHT product now allows you to extend that idea to models involving three variables. Unfortunately, it can't automatically plot surfaces through the data in a manner analogous to PROC G3D filling a line. This paper demonstrates some simple programming techniques for generating a response surface, displaying the surface and your scatterplot at the same time to examine fit (see Figure 1), and controlling the visibility of data and/or surface via a mouse. A macro is presented to automate multiple regression.

The basic concepts

The key concept in enabling INSIGHT to display a response surface is generating the surface separately and then combining it with the original data set. Assume that you've already run PROC REG and have the regression equation. You could use it to generate the surface with the commands:

```
DATA CARS;
  INPUT MPG HPOWER WEIGHT;
CARDS;
```

```
DATA GRID:
  DO HPOWER=50 TO 175 BY 25;
    DO WEIGHT=1 TO 5 BY 1;
      MPG=48.94-.067*HPOWER-6.06*WEIGHT;
      OUTPUT;
    END;
  END;
```

```
DATA SPIN;
  SET CARS GRID;
```

The increments of the DO loops (e.g., BY 25) are, of course arbitrary, and you can make them smaller numbers to generate a more dense surface, or larger numbers for a sparser surface. Keep in mind that the computer has to spin all of the points, so a dense surface may slow down spinning considerably. A dense surface also tends to obscure points which are behind the surface.

The grid data set contains the predicted surface and you can plot it with INSIGHT to yield a “spinnable” version of PROC G3D. In other words, the prediction surface is displayed, but the scatter is not.

You can also combine the grid data with the original data set so that you can view the surface and the original data at the same time:

```
DATA SPIN;
  SET CARS GRID;
```

When you first display the points with INSIGHT's Rotating Plot, a “postage stamp” may seem to appear in the display. This results from viewing the surface from directly “above” making it look like a flat grid regardless of whether it has curves in it or not. The fact that INSIGHT sets the size of the initial display at about postage stamp size doesn't help the matter! As you increase the size of the window and begin spinning the data, this will be clarified.

Controlling selection of points

It can be difficult to discriminate between data and surface grid without the ability to control colors or markers. You can add such control by creating a variable called “switch” that has a value of “data” for the original points and “grid” for the predicted surface data set:

```
DATA CARS;
  INPUT MPG HPOWER WEIGHT;
  SWITCH='DATA'; ...
DATA GRID;
  SWITCH='GRID';
  DO HPOWER=50 TO 175 BY 25;
    DO WEIGHT=1 TO 5 BY 1;
      MPG=48.94-.067*HPOWER-6.06*WEIGHT;
      OUTPUT;
    END;
  END;
DATA SPIN;
  SET CARS GRID;
```

The resulting data set combines the original variables' values, the predicted values and the switch variable indicating which is which (see Figure 2).
In order to manipulate the data and the grid separately, you must create a bar chart using the variable switch. This will result in a chart with two bars, one showing the number of observations in the original data set, and the other showing the number of points required to make up the prediction surface (see Figure 3.)

Figure 2. Data set containing original data (top three lines) and predicted response surface data (bottom three lines). Value of the variable SWITCH identifies each.

In order to manipulate the data and the grid separately, you must create a bar chart using the variable switch. This will result in a chart with two bars, one showing the number of observations in the original data set, and the other showing the number of points required to make up the prediction surface (see Figure 3.)

By choosing the bar representing the data, you will highlight those points in the rotating plot, making them easier to see. You can further enhance visibility by changing the markers or colors of the data points to something that stands out. Changing the size of the points can also greatly enhance visibility. I've found that on my 16" screen, a point size of 2 makes the points easily visible without being so large that they obscure one another.

Figure 3. Bar chart of switch variable that allows selection of the original data or predicted surface grid shown in figure 1. Darkness of the data bar indicates that the original data has been selected.

For our purposes, the height of each bar is irrelevant, you just need the ability to choose each set of points separately. By pointing at one bar or the other, you can choose whichever set of points you wish to manipulate. If there are thousands of data points, the height of the window displaying the bar chart in order to make the grid bar easier to select with the mouse.

A quicker way to achieve a similar result is to use the color palette to make the chosen points black. The black points become invisible on the black background. At first, some of the black points may obscure some of the points you're trying to see, but when you begin rotation the obscured points will appear. The points may rotate more slowly using this method because INSIGHT must still calculate the positions of the invisible points.

Try both methods and see which you like best given the speed of your workstation and the number of points you have.

Automating some steps

The method above is good for explanation, but it gets tiresome in use as you apply it to a variety of data sets. Choosing another set of variables requires you to change the ranges of the DO loops and the parameters or structure of the surface equation.

There are several approaches to automatically setting the DO loop limits. First, you can run the data through PROC MEANS, save the minimum and maximum values to an output data set and base the DO loops on them. This technique is appealing in its simplicity, but one extremely high value may generate the surface over a wider range than you prefer. Another approach is to output the mean and standard deviation to a data set and base the DO loop parameters on them. This method is less influenced by extreme values. It has the added benefit of forming the lines of the grid at standard deviation intervals about the mean, which tells you more about the variability of the data than do lines appearing at arbitrary intervals. The following example demonstrates this technique:

```
PROC MEANS MEAN STD;
VAR HPOWER WEIGHT;
OUTPUT OUT=STATS MEAN=MEAN_H MEAN_W
STD=STD_H STD_W;
DATA GRID; SET STATS;
DO HPOWER=(MEAN_H-2*STD_H) TO (MEAN_H+2*STD_H) BY (STD_H);
DO WEIGHT=(MEAN_W-2*STD_W) TO (MEAN_W+2*STD_W) BY (STD_W);
MPG...;
```
The divisors below the standard deviations are arbitrary; you can set them to whatever numbers create a pleasing looking surface. This value usually need not change from data set to data set, since the standard deviations will scale the resulting BY values appropriately.

A third approach is to run your variables through PROC STANDARD before creating the surface. This will scale your data so that the variables always have the same mean and standard deviation. This can enhance comparisons among models involving variables of different scales and allows you to use the same DO loop parameters regardless of the values of the variables:

```
PROC STANDARD DATA=CARS MEAN=0 STD=1 OUT=ZSCORES;
DATA GRID;
   DO HPOWER=-2 TO 2 BY .50;
      DO WEIGHT=-2 TO 2 BY .50;
         *This involves fewer steps and won’t change the p-values of the regression parameters, but it will change the parameters themselves.
   
Regardless of how you choose to do your loops, the parameter estimates must be calculated. In the examples above, I’ve entered them into the equation manually. You can automate this step too. The REG procedure has an OUTEST option which creates a data set containing the equation of the surface with the parameters named after the original variables and the intercept called simply, “intercept.” Since the parameters have the same names as the original variables, you must rename them before combining them with the GRID data set because it contains generated variables with those names. The OUTEST data set contains only one observation, which you can access with the SET command before running through the loops:

```
PROC REG OUTEST=EST;
   MODEL MPG=BPOWER WEIGHT;
   DATA EST;
      SET EST;
      RENAME BPOWER=B1 WEIGHT=B2;
   DATA GRID: SET EST;
      DO HPOWER=-2 TO 175 BY 25;
         DO WEIGHT=1 TO 5 BY 1;
            MPG=INTERCEPT+B1*HPOWER+B2*WEIGHT;
   
Finally, the size of the data set may be a problem. The Rotating Plot can only spin a limited number of observations; the faster your workstation, the more points you can spin. One way of dealing with this problem is to base the parameter estimates on all of your data, but display only a random sample of it. You can do this using the UNIFORM function to generate a random number between zero and one.

```
DATA CARS;
   SET CARS;
   IF UNIFORM(0) <= .50;
```

Putting it all together

You can save the steps above and use them in new programs by using the CHANGE command in the SAS editor (or your favorite editor’s global replace) to change the names of the variables and the data set being used. This method works well enough, but to really simplify matters I’ve combined these steps in a SAS macro. Macros automate the same kind of substitution that the CHANGE command does, in a manner that’s easier to use. You can use the macro in the appendix with the command:

```
%GRID(DSN~Z,X,Y,PERCENT=100);
```

If you’ve followed the explanation above, you’ll probably be able to see that the macro is nothing more than the summarization of all the steps described, with the SAS data set called &DSN, the variable predicted called &Z, and so on. The value you supply for percent will take a random sample of that percent for display, but the macro will use all of the original data in &DSN to calculate the parameter estimates.

```
%GRID(CARS,MPG,HPOWER,WEIGHT,PERCENT=100);
```

The macro substitutes the names you supply, in the order you supply them. The regression model in the macro includes second order polynomial terms and interactions, but it is easily changed to handle your model of choice.

Conclusion

The techniques described above show the various levels of automation that can be used to display a response surface to assist in examining the fit of a three dimensional regression model. I hope that you view these techniques as the tip of the iceberg. They can readily be extended to a wide variety of models and displays.
Appendix

/* A Macro to read three variables from a SAS data set, and generate a response surface grid to add to the original data for use with SAS/INSIGHT’s Rotating Plot. An example of using this might be:
%grid(car,mpg,hpower,weight,percent=50);
where car is the name of the SAS data set, mpg is being predicted by hpower and weight, and 50% of the data will be viewed in the Rotating Plot. After execution, invoke SAS/INSIGHT, open the data set called SPIN and make a rotating plot of the regression variables. Then make a bar chart of the variable SWITCH and choose one bar or the other to highlight the prediction grid or the data. Then choose colors, markers etc. to improve the view. */

%macro grid(dsn,z,x,y,percent~lOO);
  title1 "Prediction of &z by &x and &y from data &dsn";
  title2 "From data &dsn";
  /* Output means and stds to data set to use in calculating do loop limits */
  proc means mean std data=&dsn;
    var &x &y;
    output out=stats
      mean=mean x mean-y
      std=std_x-std-y;
  /* Create interaction and polynomial terms for PROC REG */
  data orig xyz;
    set &dsn;
    xy &x * &y;
    x2 &x**2;
    y2 &y**2;
    x2y2 x2 * y2;
    keep &x &y &z xy x2 y2 x2y2;
  /* Do regression and save parameter estimates to EST data set */
  proc reg data=orig_xyz outest=est;
    model &z=&x &y xy x2 y2 x2y2;
    data est: set est;
      rename intercept=b0 &x=b1 &y=b2
        xy=b3 x2=b4 y2=b5 x2y2=b6;
  /* Merge STATS and EST data sets */
  data params;
    merge est(keep=b0-b6)
      stats(drops=freq _type_); run;

  /* Generate response surface grid using regression parameters and mean, std. */
  data grid; set params;
    do x=(mean_x-2*std_x) by std_x/2;
      do y=(mean_y-2*std_y) by std_y/2;
        xy = x*y;
        x2 = x**2;
        y2 = y**2;
        x2y2 = x2 * y2;
        &z=b0 + b1*x + b2*y + b3*xy + b4*x2 + b5*y2 + b6*x2y2;
        switch='Grid'; output;
      end;
    end;
  end;
  /* Add the SWITCH variable to original data set and take a random sample to make spinning faster for large data sets */
  data orig_xyz;
    set orig - xyz;
    switch='Data';
    if uniform(0) <= &percent/100; run;
  /* Combine original &x, &y, &z variables with predicted &x, &y, and &z. */
  data spin;
    set orig_xyz grid;
    keep switch &x &y &z; run;
  %mend grid; /* this ends the macro */

/* set up your libname below */
libname save I.-;
/* mprint options shows you the full saslog expanding the macro code */
options mprint;
/* fill in your data set and variables below. After including this file and executing once, you'll need to repeat only the %grid below to try other variables */
%grid(save.car,mpg,weight,hpower,percent=100);

SAS/INSIGHT is a registered trademark of SAS Institute Inc. in the USA and other countries. TM indicates USA registration.