Applied SAS/GRAPH - Putting it all together

Regression Analysis

Graphic output is essential at all stages of regression analysis. Initially, a scatter plot of the response versus the regressor variables is desired. This can give you an idea about what type of model may be appropriate, e.g. linear, quadratic, nonlinear, etc. For example, if the data measures photosynthesis as a function of irradiance, an initial plot would be:

```
DATA IRRAD;
     <DATA STEP STATEMENTS>;

SYMBOL1 I=NONE V=DIAMOND C=BLACK;

PROC GPLOT;
     PLOT PHOTO*IRRAD=1;
```

Suppose that we decide that a linear model should be used. We can then use PROC REG to fit the linear model, output the usual regression diagnostics and create some useful plots. These will be used for plotting.

```
PROC REG;
     MODEL PHOTO = IRRAD;
     OUTPUT OUT=PRED P=YHAT RSTUDENT = RESID
                    L95 = LOW U95 = UP;

     SYMBOL1 I=NONE V=DIAMOND C=BLACK;
     SYMBOL2 I=JOIN V=NONE C=BLUE L=1;
     SYMBOL3 I=JOIN V=NONE C=GREEN L=3;

PROC GPLOT;
     PLOT PHOTO*IRRAD=1 YHAT*IRRAD=2
          LOW*IRRAD=3 UP*IRRAD=3 / OVERLAY;
     PLOT RESID*IRRAD=1 / VREF=0;
```

These statements do several things. First, PROC REG outputs the regression results into a data set called PRED. This data set has variables YHAT (predicted photosynthesis), RESID (studentized residuals), and LOW & UP (lower and upper prediction bounds on the line). Three symbols are then defined. The first is for black diamonds, the second is for joined points - solid blue line and the last is for joined points - dashed green lines. All this is put into PROC GPLOT to produce a plot with the data (black diamonds) overlayed with the estimated model (blue line) and it’s 95% prediction bounds (green dashed lines). The second plot then shows the residuals versus IRRAD as black diamonds. A vertical reference line at 0 is also drawn on this plot to make interpretation easier.

Similar programs can be used with multiple regression. In this case, PROC G3D may be added also to view the resulting response surface.

```
PROC REG;
    MODEL DURATION = INTEN TEMP INTEN*TEMP;
    OUTPUT OUT=PRED P=YHAT RSTUDENT = RESID;

    SYMBOL1 I=NONE V=DIAMOND C=BLACK;

    PROC GPLOT;
    PLOT RESID*INTEN=1 RESID*TEMP=1 / VREF=0;

    PROC G3D;
    PLOT INTEN*TEMP=YHAT/CTOP=BLUE CBOTTOM=GREEN;
```

This time we need to plot two sets of residuals, but no predicted line because this is a surface. To see the surface we use G3D with the plot statement. It is drawn in blue with a green bottom. It is not possible to display the data points on this plot with the surface and this would likely be too messy to interpret anyway. We have to rely on the residual plots to visualize the model fit. This is especially true when more than two variables are included in the model.

Graphic display in nonlinear model fitting is even more important. Nonlinear estimation is an approximate process and it is critical to examine the fitted model against the real data. Statements for doing this are similar to PROC REG. The example below fits a cumulative logistic model to seed germination over time using the Gauss-Newton algorithm (See NLIN Workshop, Handout #3).

```
PROC NLIN METHOD=GAUSS;
     PARMS M=100 L=11 B=1;
     MODEL GERM=M/(1+EXP(_B*(TIME_L)));
     DER.M =1/(1+EXP(_B*(TIME_L)));
     DER.L=_(M*(B*EXP(_(B*(TIME_L))))/
     (1+EXP(_(B*(TIME_L))))**2);
     DER.B=_(M*(_((TIME_L)*EXP(_(B*(TIME_L)))))/
     (1+EXP(_(B*(TIME_L))))**2);
     OUTPUT OUT=PRED1 P=PR STUDENT=RESID;
PROC SORT;
     BY TIME;
SYMBOL1 V=DIAMOND I=NONE C=BLACK;
SYMBOL2 V=NONE I=JOIN W=2 C=ORANGE;

PROC GPLOT;
     PLOT GERM*TIME=1 PR*TIME=2/OVERLAY;
     PLOT RESID*TIME=1/VREF=0;
```

Here, the data has been sorted before plotting. Since we are representing a curve by joining points together with a straight line, we must make sure all the points are in ascending order before plotting.

Finally, it is occasionally of interest to plot two regression lines together, e.g. regression comparison or dummy variable regression (REG Workshop, Handout #4). We can use the third variable technique together with the symbol statement to accomplish this.

```
PROC GLM;
     CLASS VAR;
     MODEL YIELD = VAR VAR*PH/SOLUTION NOINT;

     OUTPUT OUT=PRED P=YHAT;

     CONTRAST 'Intercepts' VAR 1 _1;
     CONTRAST 'Slopes' VAR*PH 1 _1;
     CONTRAST 'Lines'  VAR 1 _1,
                       VAR*PH 1 _1;

     SYMBOL1 I=JOIN V=NONE C=YELLOW;
     SYMBOL2 I=JOIN V=NONE C=ORANGE;

     PROC GPLOT;
          PLOT YHAT*PH = VAR;
```

This gives a plot with two lines, yellow and orange, representing the two varieties. This could also be accomplished with PROC REG and the BY statement:

```
PROC REG;
     MODEL YIELD = PH;
     BY VAR;
     OUTPUT OUT=PRED P=YHAT;

SYMBOL1 I=JOIN V=NONE C=YELLOW;
SYMBOL2 I=JOIN V=NONE C=ORANGE;

PROC GPLOT;
     PLOT YHAT*PH = VAR;
```

Analysis of Variance

In Analysis of Variance (ANOVA) it is often important to plot mean values to examine treatment and interaction effects. For treatment main effects, procedures GPLOT or GCHART could be used. Using an example of a variety - fertilizer trial (GLM Workshop, Handout #2):

```
PROC GLM;
     CLASS VAR FERT;
     MODEL YIELD = VAR FERT VAR*FERT;

     MEANS VAR FERT/LSD;
     LSMEANS VAR FERT/ OUT=MEANS;

PROC GCHART;
     VBAR VAR / SUMVAR=LSMEAN;

PROC GCHART;
     VBAR FERT / SUMVAR=LSMEAN;
```

Note that the LSMEANS statement is required here. The MEANS statement can not output values to a data set. If the data is unbalanced or Analysis of Covariance is used, LSMEANS are the appropriate choice anyway. With balanced data, MEANS and LSMEANS will produce identical values.

When interactions are present, an interaction plot can be made using GPLOT;

```
PROC GLM;
     CLASS VAR FERT;
     MODEL YIELD = VAR FERT VAR*FERT;

     LSMEANS VAR*FERT/ OUT=MEANS;

SYMBOL1 I=JOIN V=SQUARE C=ORANGE;
SYMBOL2 I=JOIN V=CIRCLE C=BLACK;

PROC SORT;
     BY VAR FERT;

PROC GPLOT;
     PLOT LSMEAN*FERT=VAR;
```

This gives two line plots, one for each variety. The plots depict the mean value versus fertilizer level. In order for these plots to work, you must define one SYMBOL statement for each level of the coding variable, VAR in this case. Since you can only define 10 symbols, you are limited to 10 levels of treatment, but experimental designs with this many levels would be of questionable value anyway.

Categorical Analysis

Categorical analysis are often visualized using frequency or percentage charts (CATMOD Workshop). The obvious choice of procedures here is GCHART.

```
PROC GCHART;
     VBAR FIELD / FREQ=COUNT;
     VBAR DEGREE / FREQ=COUNT;
```

This produces two bar charts, one for field of study and the other for degree obtained. The height of the bars indicates number or frequency of response in each category. The FREQ= option is used because the data has already been summarized into counts for each level of response, e.g. the variable COUNT. These might also be displayed in the form of pie charts:

```
PROC GCHART;
     PIE FIELD / FREQ=COUNT FILL=S;
     PIE DEGREE / FREQ=COUNT FILL=X;
```

The first chart uses solid colors to fill the slices, while the second uses cross-hatching.

Percentages can also be done:

```
PROC GCHART;
     VBAR FIELD / FREQ=COUNT PERCENT;
     VBAR DEGREE / FREQ=COUNT CPERCENT;
```

The first line produces bar height as a percentage of the total count whereas the second line produces bar height as a cumulative percentage of bar height.

It is possible to display “three dimensional” bar charts in SAS using the BLOCK statement within GCHART, however this is not recommended. Such charts can be confusing to interpret and, in fact, may be misleading. The goal of all graphic displays is to clarify the ideas being presented and not to obscure them.