Examples Using the PLS Procedure

Table of Contents

Examples Using the PLS Procedure

The examples in this report use the experimental PLS procedure in SAS/STAT software, Release 6.12, to model data by partial least squares (PLS) regression. A system of macros is used with PROC PLS to produce high-resolution plots for the model.

Example 1. Predicting Biological Activity

Introduction

The following example, from Umetrics (1995), demonstrates the use of partial least squares in drug discovery. New drugs are developed from chemicals that are biologically active. Testing a compound for biological activity is an expensive procedure, so it would be useful to be able to predict biological activity from other cheaper chemical measurements. In fact, computational chemistry makes it possible to calculate certain chemical measurements without even making the compound. These measurements include size, lipophilicity, and polarity at various sites on the molecule. The SAS statements to create a SAS data set named PENTA containing these data are given in Appendix 1.

You would like to study the relationship between these measurements and the activity of the compound, represented by the logarithm of the relative Bradykinin activating activity (**log_RAI**). Notice that these data consist of many predictors relative to the number of observations. Partial least squares is especially appropriate in this situation as a useful tool for finding a few underlying predictors that account for most of the variation in the response. Typically, the model is fit for part of the data (the training set), and the quality of the fit is judged by how well it predicts the other part of the data (the prediction set). For this example, the first fifteen observations serve as the training set and the rest constitute the test set (refer to Ufkes et al. 1978, 1982).

First PLS Model

When you fit a PLS model, you hope to find a few PLS factors (also known as *components* or *latent variables*) that explain most of the variation in both predictors and responses. Factors that explain response variation well provide good predictive models for new responses, and factors that explain predictor variation well are well represented by the observed values of the predictors. The following statements set the macro variables for this example and then fits a PLS model with two components. Appendix 2 lists the macros called in these examples.

```
   /*********************************************************/
   /  Select the first 15 observations for the training set  /
   /  from the original data set, PENTA.                     /
   /*********************************************************/

data penta_a; set penta;
   if _N_ <= 15;
      n=_N_;

   /*********************************************************/
   /  Set Parameters for Macros                              /
   /*********************************************************/

%global xvars yvars predname resname xscrname yscrname
        num_x num_y lv;

   %let xvars=S1 L1 P1 S2 L2 P2 S3 L3 P3 S4 L4 P4 S5 L5 P5;
   %let yvars=log_RAI;
   %let ypred=yhat1;
   %let yres=yres1;
   %let predname=yhat;
   %let resname=res;
   %let xscrname=xscr;
   %let yscrname=yscr;
   %let num_y=1;
   %let num_x=15;

   /*********************************************************/
   / Fit the PLS model                                       /
   /*********************************************************/

proc pls data=penta_a method=pls outmodel=est1 lv=2;
   model &yvars = &xvars;
   output out=outpls p=yhat1 yresidual=yres1
          xresidual=xres1-xres15 xscore=xscr yscore=yscr
          stdy=stdy stdx=stdx h=h press=press t2=t2
          xqres=xqres yqres=yqres;
run;

   %let lv=2;                *** Used 2 PLS components ***;
```

The PLS procedure displays a table, shown in Output 1.1, showing how much

predictor and response variation is explained by each PLS component.

Output 1.1. Amount of Training Set Variation Explained

The PLS Procedure
Percent Variation Accounted For

Number of Latent Variables	Model Effects Current	Model Effects Total	Dependent Variables Current	Dependent Variables Total
1	16.9014	16.9014	89.6399	89.6399
2	12.7721	29.6735	7.8368	97.4767

The procedure also produces two data sets: the EST1 data set containing information on the model fit, and the OUTPLS data set containing predictions, residuals, scores, and other information.

From Output 1.1, note that 97% of the response variation is already explained, but only 29% of the predictor variation is explained.

The PLS model has the form

$$X = TP' + E, \text{ and}$$
$$Y = UQ' + F$$

where X and Y are the matrices of predictors and responses. The matrices on the right-hand side of this model are defined by

T	=	X-scores	U	=	Y-scores
P	=	X-loadings	Q	=	Y-loadings
E	=	X-residuals	F	=	Y-residuals

Partial least squares algorithms choose successive orthogonal factors that maximize the covariance between each X-score and the corresponding Y-score. For a good PLS model, the first few factors show a high correlation between the X- and Y-scores. The correlation usually decreases from one factor to the next. You can plot the X-scores T versus the corresponding Y-scores U using the following macro call.

```
%plot_scr(outpls);
```

The plots for these data appear in Figures 2 and 3. The numbers on the plot represent the observation number in the PENTAPEP data set, which appears in Appendix 1. For this example, the figures show high correlation between X- and Y-scores for the first component but somewhat looser correlation for the second component.

You can also plot the X-scores against each other to look for irregularities in the data. You should look out for patterns or clearly grouped observations. If you see a curved pattern, for example, you may want to add a quadratic term. Two or more groupings of observations indicate that it might be better to analyze the groups separately. The following macro call produces plots of scores for consecutive PLS components for as many components as desired, up to the number of components fit.

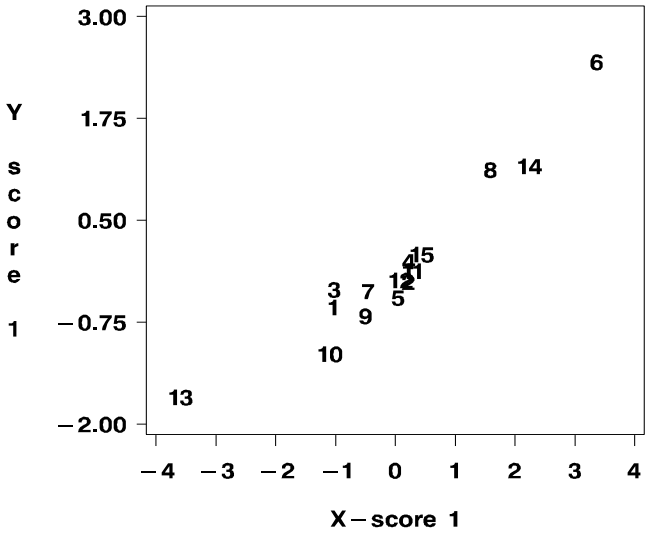

Figure 2: First X- and Y-scores for Penta-Peptide Model 1

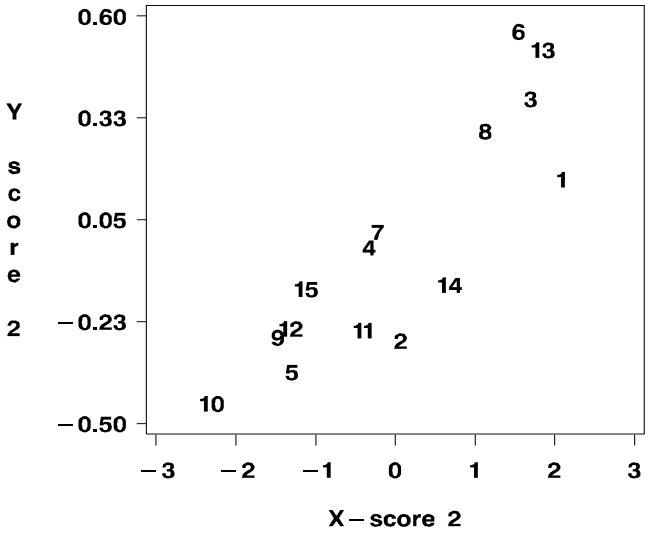

Figure 3: Second X- and Y-scores for Penta-Peptide Model 1

```
%plotxscr(outpls,max_lv=2);
```

The plot of the first and second X-scores is shown in Figure 4. This plot appears to show most of the observations close together, with a few being more spread out with larger positive X-scores for component 2. Observation 13 stands out the most and has been the most extreme on all three plots so far. This run may be influential in the PLS analysis, and thus you should check to make sure it is reliable. There are not any distinct grouping patterns.

Plots of the weights give the directions toward which each PLS factor projects. They show which predictors are most represented in each factor. Those predictors with small weights are less important than those with large weights in absolute value.

The X-weights W represent the correlation between the X-variables and the Y-scores

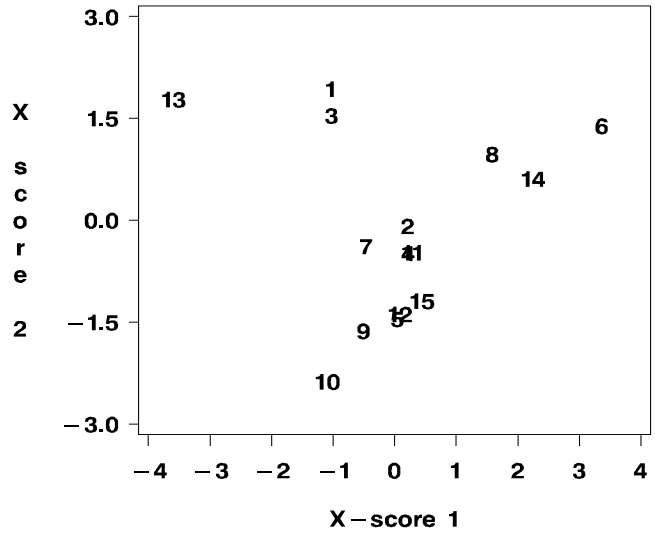

Figure 4: First and Second X-scores for Penta-Peptide Model 1

U. The Y-loadings Q represent the correlation between the Y-variables and the X-scores T. The X-loadings P represent the directions of the lines $\mathbf{u} = \mathbf{b}'\mathbf{t}$ in the X-space. The X-loadings and X-weights are usually very similar to each other.

You can produce these plots with the following macro calls.

```
   /*********************************************************/
   /  Compute the X-weights for each PLS component          /
   /*********************************************************/

%get_wts(est1,dsxwts=xwts);

   /*********************************************************/
   /  Plot X-weights w1 and w2 for the two components       /
   /*********************************************************/

%plot_wt(xwts,max_lv=2);

   /*********************************************************/
   /  Compute X-loadings p1-p2 for the two components       /
   /*********************************************************/

%getxload(est1,dsxload=xloads);

   /*********************************************************/
   /  Plot X-loadings p1 and p2 for the two components      /
   /*********************************************************/

%pltxload(xloads,max_lv=2);
```

The plot of the X-weights is shown in Figure 5. The plot of the X-loadings, which is similar, is not shown.

The weights plot shows a cluster of X-variables that are weighted at nearly zero for

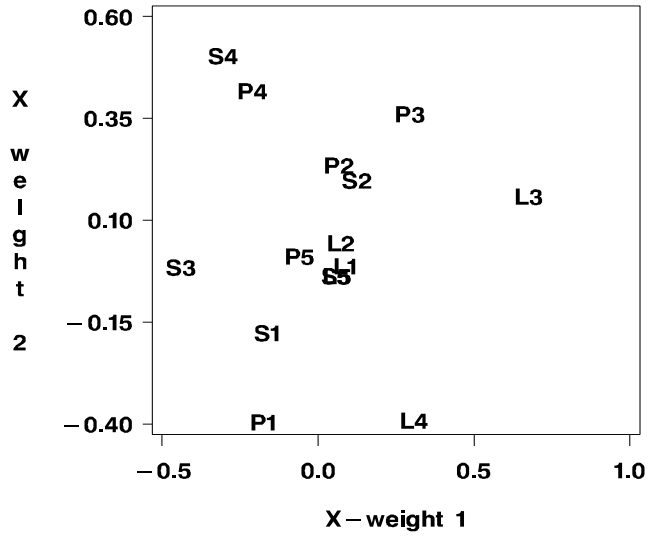

Figure 5: First and Second X-weights for Penta-Peptide Model 1

both components. These variables add little to the model fit and removing them may improve the model's predictive capability.

Residual plots and normal quantile plots help in detecting outliers that might be harming the fit; these plots also help in detecting nonnormality, autocorrelations, and heteroscedasticity, all of which can cause various problems in constructing confidence and tolerance bounds for predictions. The ideal residual plot looks like a rectangular point cloud with a majority of the points falling in the vertical middle third of the plot. In an ideal normal plot, the points fall on a straight line. You can produce the plot of residuals versus predicted values with the **`%res_plot`** macro and the normal quantile plot of the residuals with the **`%nor_plot`** macro for each response variable.

```
%res_plot(outpls);

%nor_plot(outpls);
```

The resulting plots appear in Figures 6 and 7.

For these data, the plot of residuals versus predicted values in Figure 6 shows nothing unusual, but the normal quantile plot in Figure 7 shows that several observations are more extreme at the lower end than what you would expect under normality.

To determine which factors to eliminate from the analysis, you can look at the regression coefficients in the B(PLS) matrix (which in this case is a column vector) and at the Variable Importance for the Projection (VIP) of each factor. The regression coefficients represent the importance each factor has in the prediction of the response. The VIP represents the value of each predictor in fitting the PLS model for both predictors and responses. If a predictor has a relatively small coefficient (in absolute value) *and* a small value of VIP (Wold (1994) considers less than 0.8 to be ''small''), then it is a prime candidate for deletion. The following statements produce coefficients and the VIP.

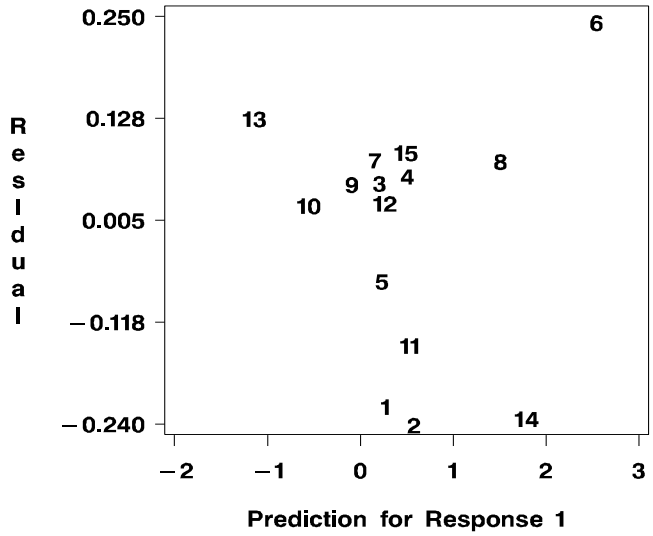

Figure 6: Residuals vs. Predicted Values for Penta-Peptide Model 1

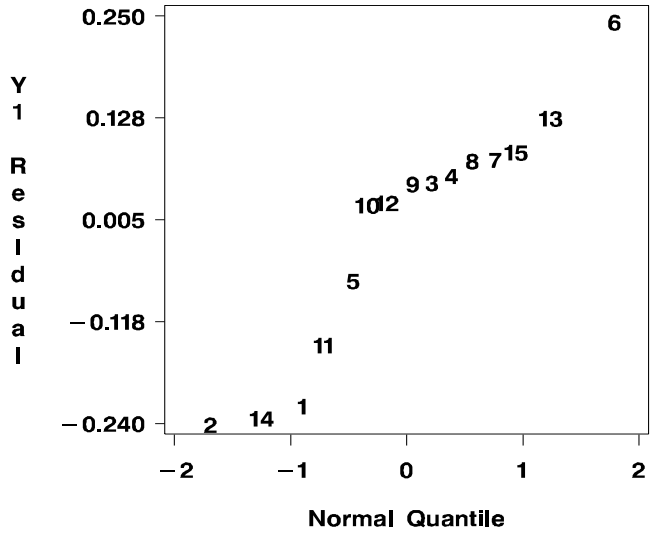

Figure 7: Normal Quantile Plot of Residuals for Penta-Peptide Model 1

```
%get_bpls(est1,dsout=bpls);

%get_vip(est1,dsvip=vip_data);

data eval;
   merge bpls vip_data;
run;

proc print data=eval;
run;
```

The output appears in Output 1.2.

Output 1.2. Estimated PLS Regression Coefficients and VIP (Model 1)

OBS	X_VAR	B1	VIP
1	S1	-0.13831	0.61108
2	L1	0.05720	0.31822
3	P1	-0.19064	0.75127
4	S2	0.12383	0.50482
5	L2	0.05909	0.27123
6	P2	0.09361	0.35927
7	S3	-0.28415	1.57775
8	L3	0.47131	2.43480
9	P3	0.26613	1.13222
10	S4	-0.09145	1.22255
11	L4	0.12265	1.17994
12	P4	-0.04878	0.88380
13	S5	0.03320	0.21288
14	L5	0.03320	0.21288
15	P5	-0.03320	0.21288

For this data set, the variables L1, L2, P2, P4, S5, L5, and P5 have small absolute coefficients and small VIP, so they are dropped from the analysis.

Looking back at the loadings plot, you can see that these variables tend to be the ones near zero for both PLS components.

Reduced Model Analysis

The statements below refit the model with the seven insignificant predictors dropped.

```
   /*********************************************************/
   /  Refit the PLS model with 7 X-variables deleted        /
   /*********************************************************/

   %let xvars=S1 P1 S2 S3 L3 P3 S4 L4;
   %let num_x=8;

proc pls data=penta_a method=pls outmodel=est1b lv=2;
   model &yvars = &xvars;
   output out=outpls1b p=yhat1 yresidual=yres1
          xresidual=xres1-xres8 xscore=xscr yscore=yscr
          stdy=stdy stdx=stdx h=h press=press t2=t2
          xqres=xqres yqres=yqres;
run;

   /*********************************************************/
   / Plot a normal quantile plot of the residuals           /
   / (for comparison to the original fit).                  /
   /*********************************************************/

%nor_plot(outpls1b);
```

```
/*********************************************************/
/ Plot the X-scores vs. Y-scores for each component.     /
/*********************************************************/

%plot_scr(outpls1b);
```

The printed output from the PLS procedure appears in Output 1.3, the normal quantile plot appears in Figure 8 and the plot of the second X and Y-scores against each other appears in Figure 9.

Output 1.3. Amount of Training Set Variation Explained (Reduced Model)

The PLS Procedure
Percent Variation Accounted For

Number of Latent Variables	Model Effects Current	Model Effects Total	Dependent Variables Current	Dependent Variables Total
1	28.5163	28.5163	90.7983	90.7983
2	19.1750	47.6913	5.9785	96.7769

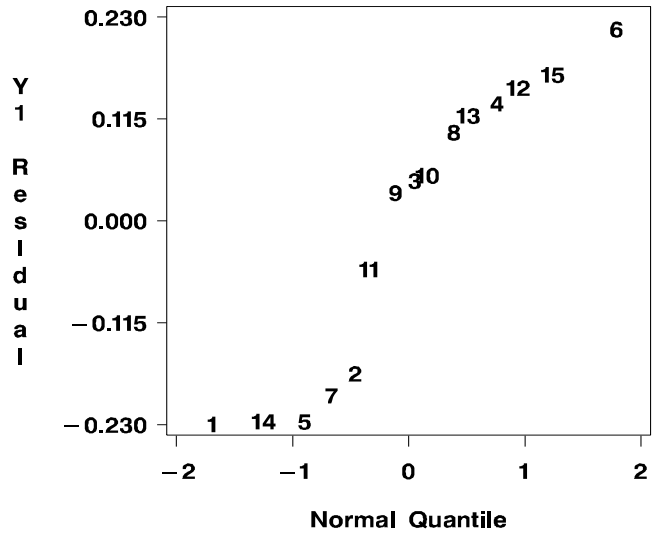

Figure 8: Normal Quantile Plot for Penta-Peptide Reduced Model

When the model is fit with the remaining eight predictors, the R-squared values for X improve to 47% for two PLS components. See Output 1.3.

So if you drop predictors that appear to be the least related to Y, you find that the PLS factors are better represented by the data in the remaining X-space. Note that the normal quantile plot (Figure 8) is closer to being linear than previously (Figure 7).

You can also see in Figure 9 that the correlation between the X- and Y-scores for the second component appears stronger.

Another way to check for outliers in the model is to look at the Euclidean distance from each point to the PLS model in both X and Y. No point should be dramatically

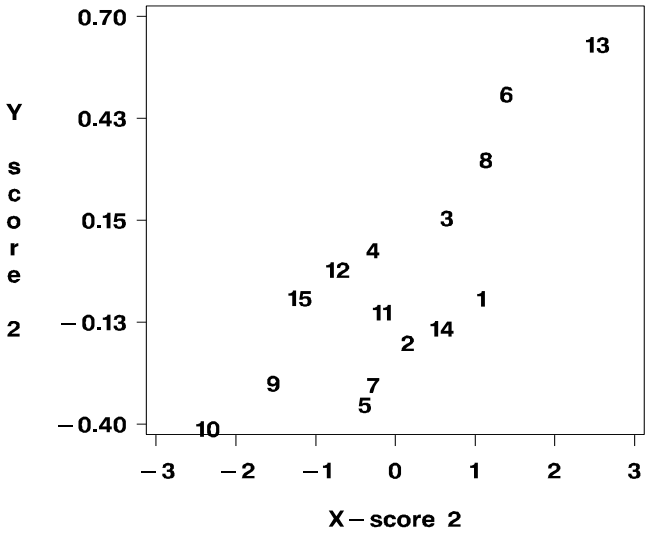

Figure 9: Second X- and Y-scores for Penta-Peptide Reduced Model

farther from the model than the rest. If there is a group of points that are all farther from the model than the rest, it may be that they have something in common and should be analyzed separately. The following statements compute and plot these distances to the model, which Umetrics (1995) call DModX and DModY.

```
%get_dmod(outpls1b,dsdmod=distmd1b,qresname=qres,id=n);

proc gplot data=distmd1b;
   plot dmodx*n;
   symbol1 i=needles v=dot;
   symbol2 i=needles v=dot;
run;
```

The plots are shown in Figures 10 and 11.

There appear to be no outliers. Overall, this second model appears to be more satisfactory than the first one.

Predictions for the Remaining Observations

You can make predictions for the test set (observations 16-31 of the original data) by appending it to the training set with missing values for the responses and specifying the P= option in the OUTPUT statement. Then you can check the predictions based on the model for the first 15 observations against their actual values (except for observation 31, which is missing in the response).

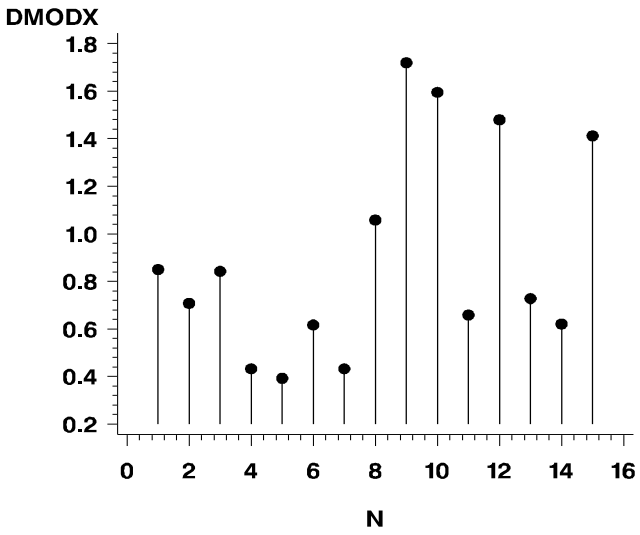

Figure 10: Distances from the X-variables to the Model (Training Set)

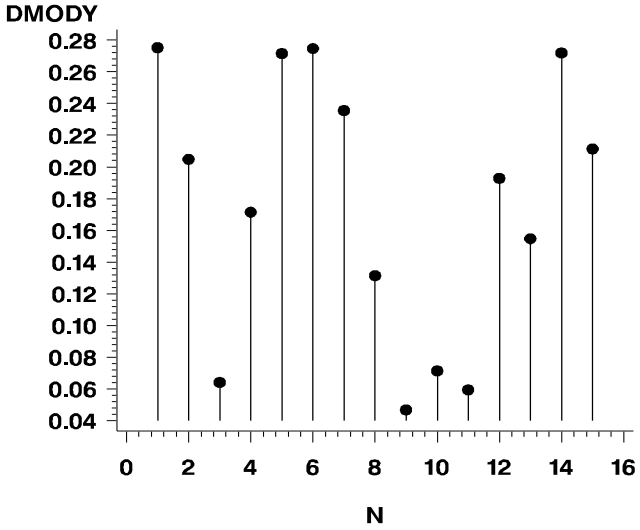

Figure 11: Distances from the Y-variables to the Model (Training Set)

```
   /*********************************************************/
   /  Refit the model with missing values at the points      /
   /  to be predicted.                                       /
   /*********************************************************/

data penta_b; set penta;
   if n > 15 then log_RAI=.;

proc pls data=penta_b method=pls lv=2 noprint;
   model &yvars = &xvars;
   output out=outpls2 p=yhat1 yresidual=yres1
          xresidual=xres1-xres8 xscore=xscr yscore=yscr
          stdy=stdy stdx=stdx h=h press=press t2=t2
          xqres=xqres yqres=yqres;
```

```
run;

   /*********************************************************/
   /   Put the predicted values and actual observations in   /
   /   the same data set.                                    /
   /*********************************************************/

data outpls2a; set outpls2(keep=&ypred); n=_N_; run;

data penta_c; set penta(keep=&yvars); n=_N_; run;

data predict; merge penta_c outpls2a; by n; run;

   /*********************************************************/
   /  Calculate the residuals at the points in the test set. /
   /*********************************************************/

data predict; set predict;
   yres1=log_RAI-yhat1;
      if _N_=31 then delete;
run;

   /*********************************************************/
   /  Compare the test set and training set residuals.       /
   /*********************************************************/

%res_plot(predict);
```

Figure 12 displays the plot. You can also print out the predictions in the PREDICT data set, but these are not displayed here.

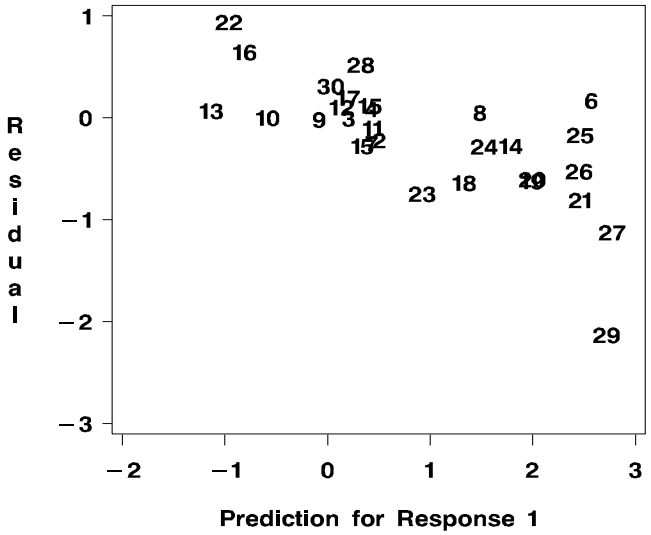

Figure 12: Residuals for all Observations Based on Model for Training Set

In Figure 12, the residuals for observations 16-30 calculated based on predictions from observations 1-15 appear to have a slight systematic pattern.

Observations 27 and 29 stand out the most, and in general it appears that the

model is slightly underpredicting the Y-activity when it predicts low activity and overpredicting it when it predicts high activity.

To see if the new observations are representative of the model for X, you can call the **%get_dmod** macro again and plot the distances.

```
%get_dmod(outpls2,dsdmod=distmod2,qresname=qres,id=n);

   axis1 order=(0 to 32 by 8);

proc gplot data=distmod2;
   plot dmodx*n/haxis=axis1;
   symbol1 i=needles v=dot;
run;
```

The plot appears in Figure 13.

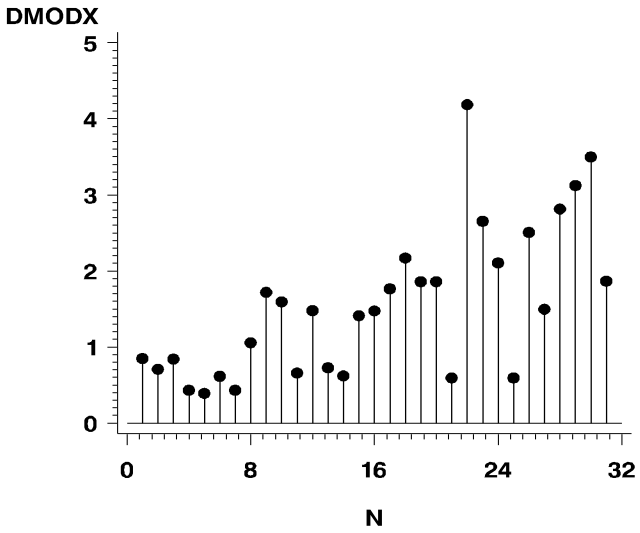

Figure 13: Distances from the X-variables to the Model (All Data)

In Figure 13, the distances of observations 16-30 to the PLS model for the predictors are much larger on average than the distances for the first 15 observations.

This indicates that the X-values for the first 15 are not as representative of the second 15 as you would like and it helps explain the problems in prediction.

Conclusion

In this example, partial least squares provided an effective method for predicting the chemical activity of a penta-peptide by taking only eight total measurements of size, lipophilicity, or polarity. Two underlying factors based on these quantities accounted for almost all of the variation in the response and provided a good model for predicting responses in the prediction set.

Example 2. Spectrometric Calibration (Observed Data)

Introduction

Spectrometric calibration is another type of problem where partial least squares is very effective in predicting responses from a large number of predictors. As described in Tobias (1995), to calibrate an instrument you run compounds of known composition through the spectrograph and observe the spectra they yield. Based on this data, you fit a model that you then use to predict concentrations of unknown samples based on their spectra. The next two examples come from calibration problems.

In the MSWKAL data set, again supplied by Umetrics (1995), researchers would like to fit a spectrographic model so they can determine the amounts of three compounds present in samples from the Baltic Sea: LS (lignin sulfonate: pulp industry pollution), HA (humic acids: natural forest products), and DT (optical whitener from detergent). The data set consists of 16 samples of known concentrations of LS, HA, and DT, with spectra based on 27 frequencies (or equivalently, wavelengths), as well as two samples of known concentration for use in checking the robustness of the model. (Refer to Lindberg et al. 1983.)

The statements to create a SAS data set named MSWKAL for these data are supplied in Appendix 1.

First Model Fit

To isolate a few underlying spectral factors that provide a good predictive model, you can fit a PLS model to the 16 samples. To choose the number of PLS components you use some form of cross-validation. In cross-validation, the data set is divided into two or more groups. You fit the model to all groups but one, then check the capability of the model to predict responses for the group left out. Repeating this process for each group, you then can measure the overall capability of a given form of the model. The Predicted REsidual Sum of Squares (PRESS) statistic is based on the residuals generated by this process. You can choose the number of PLS components based on the model with the minimum PRESS statistic or based on a hypothesis test such as one that uses the PRESS statistic for each model. In this cross-validation test approach, the PLS procedure with the CVTEST(STAT=PRESS) option selects the smallest model that has a PRESS statistic insignificantly larger than the absolute minimum PRESS statistic.

One important issue is selection of the number and composition of groups to leave out when doing cross-validation. Umetrics (1995) recommends having seven or more groups. Shao (1993) recommends against using ordinary cross-validation with groups of size one. For this data set, eight groups of size two should work well.

In the PLS procedure, you can accomplish this group selection by using the CV=SPLIT(8) option to choose eight cross-validation groups composed of observa-

tions 1 and 9, observations 2 and 10, and so on.

The following statements set the macro variables for this data set and fit the first PLS model using the preceding criteria. The macros used in this example are listed in Appendix 2.

```
data mswkal_a; set mswkal;
   if n <= 16;

%global xvars yvars predname resname xscrname yscrname
        num_x num_y lv;

   %let title1=PLS Analysis of Umetrics MSWKAL Example;
   %let xvars=v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12 v13 v14
        v15 v16 v17 v18 v19 v20 v21 v22 v23 v24 v25 v26 v27;
   %let yvars=ls ha dt;
   %let ypred=yhat1 yhat2 yhat3;
   %let yres=yres1 yres2 yres3;
   %let predname=yhat;
   %let resname=res;
   %let xscrname=xscr;
   %let yscrname=yscr;
   %let num_y=3;
   %let num_x=27;

proc pls data=mswkal_a method=pls outmodel=est1 cv=split(8)
         cvtest(stat=press);
   model ls ha dt = v1-v27;
   output out=outpls p=yhat1-yhat3 yresidual=yres1-yres3
          xresidual=xres1-xres27 xscore=xscr yscore=yscr
          stdy=stdy stdx=stdx h=h press=press t2=t2
          xqres=xqres yqres=yqres;
run;

   %let lv=2;        *** Number of PLS components in model ***;
```

The PLS output appears in Output 2.1 and Output 2.2.

Output 2.1. Cross-Validation Results

1

The PLS Procedure
Cross Validation for the Number of Latent Variables

Number of Latent Variables	Root Mean PRESS	Test for larger residuals than minimum: Prob > PRESS
0	1.0975	0.00100
1	0.8806	0
2	0.8305	0.2100
3	0.6289	0.1700
4	0.5792	0.00400
5	0.7049	0.00200
6	0.4613	0.00400
7	0.4151	1.0000
8	0.4569	0.9620
9	0.4471	0.9190
10	0.4247	0.9710
11	0.4322	0.8700
12	0.4412	0.9180
13	0.4435	0.9510
14	0.4435	0.9510
15	0.4435	0.9510
16	0.4435	0.9510
17	0.4435	0.9510
18	0.4435	0.9510
19	0.4435	0.9510
20	0.4435	0.9510
21	0.4435	0.9510
22	0.4435	0.9510
23	0.4435	0.9510
24	0.4435	0.9510
25	0.4435	0.9510
26	0.4435	0.9510
27	0.4435	0.9510

Minimum Root Mean PRESS = 0.415117 for 7 latent variables
Smallest model with p-value > 0.1: 2 latent variables

Output 2.2. Percentages of Variation Explained by Model

2

The PLS Procedure
Percent Variation Accounted For

Number of Latent Variables	Model Effects: Current	Model Effects: Total	Dependent Variables: Current	Dependent Variables: Total
1	97.4607	97.4607	41.9155	41.9155
2	2.1830	99.6436	24.2435	66.1590

The cross-validation results in Output 2.1 show that the procedure selected a model with two PLS components (latent variables) because that is the simplest model with

a PRESS statistic that is insignificantly different from the absolute minimum PRESS value. Output 2.2 shows that the PLS model explains more than 99% of the variation in predictors and about 66% of the variation in responses. If you had not used the CVTEST option, the procedure would have fit a model with seven PLS components instead of two.

To check the quality of the model, you can check to see if the X-scores and respective Y-scores are highly correlated using the following command.

```
%plot_scr(outpls);
```

The plots appear in Figures 14 and 15. Recall that the numbers on the plots refer to the observation numbers in the MSWKAL data set, given in Appendix 1.

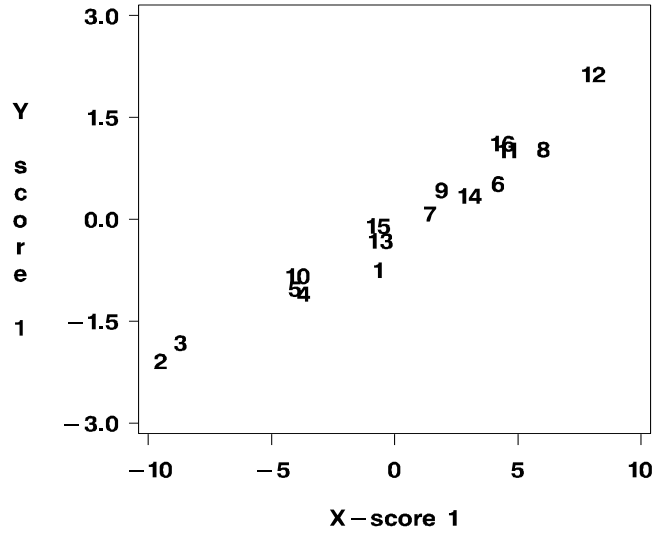

Figure 14: First X- and Y-scores for MSWKAL Model

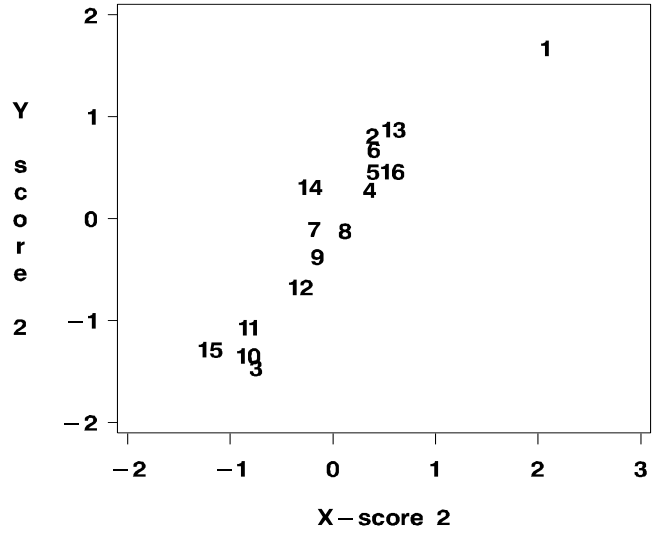

Figure 15: Second X- and Y-scores for MSWKAL Model

From these plots, you can see that the X- and Y-scores are highly correlated for the first two PLS components, indicating a good model. To check for irregularities in the predictors, such as outliers or distinct groupings, you can plot the X-scores against each other using the following statements.

```
%plotxscr(outpls);
```

The plot of the first and second X-scores is shown in Figure 16. The plot of X-scores shows no irregularities.

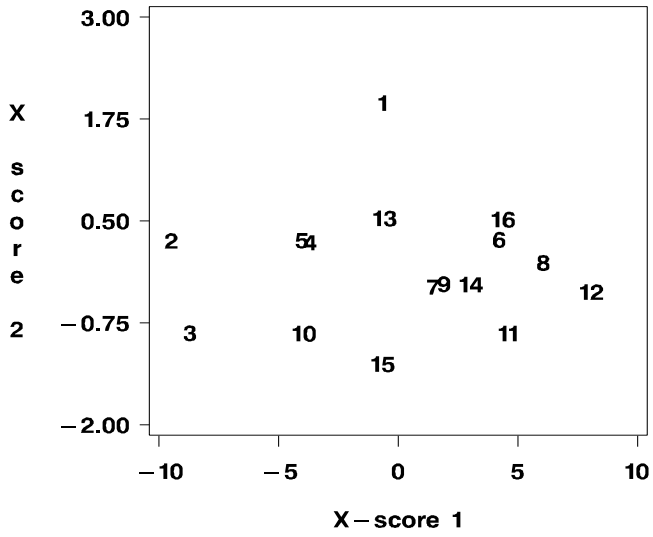

Figure 16: First and Second X-scores for MSWKAL Model

To see which predictors are most dominant in each factor, you can plot the weights and loadings across the range of predictors. Since the predictors are frequencies, it makes sense to plot the weights and loadings across frequencies rather than against each other. You can use the following statements to generate these plots.

```
   /*********************************************************/
   /  Compute the X-Weights for each PLS component          /
   /*********************************************************/

%get_wts(est1,dsxwts=xwts);

   /*********************************************************/
   /  Plot the X-weights vs. the frequency on the same axes  /
   /*********************************************************/

%pltwtfrq(xwts,plotyvar=w,plotxvar=n,max_lv=&lv,
          label=Weight);
```

```
/*********************************************************/
/  Compute X-loadings p1-p2 for the two components       /
/*********************************************************/

%getxload(est1,dsxload=xloads);

/*********************************************************/
/  Plot the X-loadings for each component vs. frequency  /
/*********************************************************/

%pltwtfrq(xloads,plotyvar=p,plotxvar=n,max_lv=&lv,
   label=Loading);
```

Figure 17 displays the weight plot across frequencies. The loadings plot looks very similar.

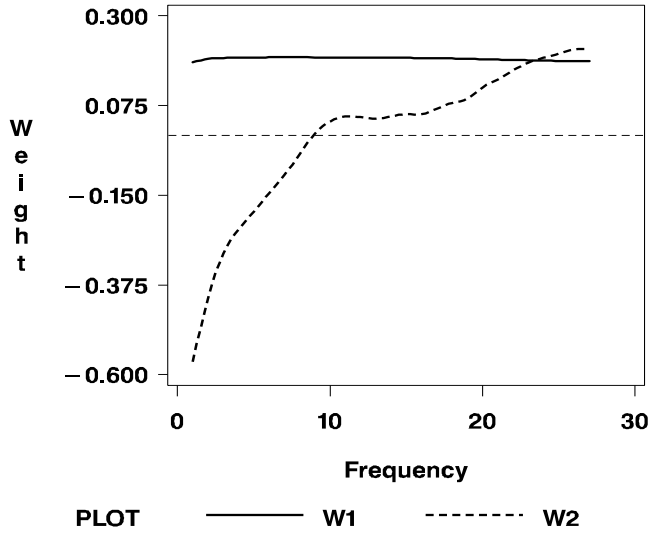

Figure 17: X-weights Across Frequencies for MSWKAL Model

The plot shows a fairly constant weight across frequencies for the first PLS component, revealing that the integral of the spectrogram is the most important predictor. For the second component, the weights increase as the frequency increases. The second component is a smoothed contrast between frequencies below and above 9 or so.

The X-loadings give the combination of predictors that comprise each PLS component. In the same way, you can examine the Y-loadings to see how each PLS component represents the responses. The following statements compute the Y-loadings and then plot them for each PLS component.

```
/*********************************************************/
/  Compute Y-loadings q1-q2 for the two components       /
/*********************************************************/

%getyload(est1,dsyload=yloads);
```

```
   /*********************************************************/
   /  Plot the Y-loadings vs. the PLS components            /
   /*********************************************************/

%plt_y_lv(est1);
```

The plot appears in Figure 18.

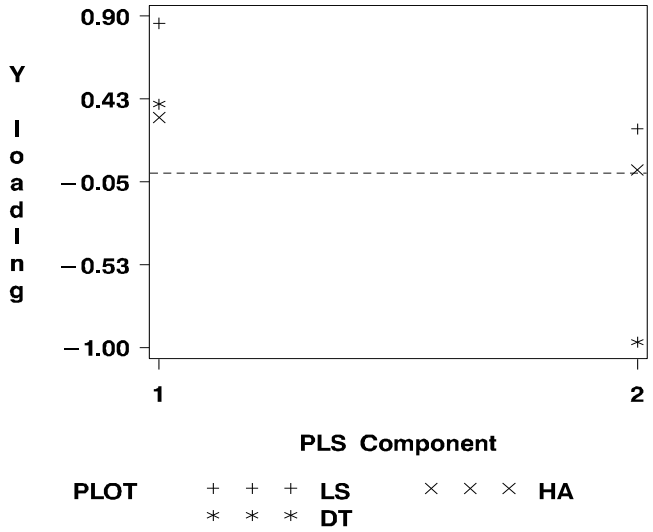

Figure 18: Y-loadings vs. PLS Component for MSWKAL Model

The plots show that the first component is based mainly on LS, with some emphasis on the other two responses. The second component emphasizes DT and, to a lesser extent, LS.

To see which frequencies are important, you can look at the B(PLS) regression coefficient matrix and at the Variable Importance for the Projection (VIP). Since the predictors are ordered, it makes sense to plot VIP and B(PLS) against them. It also may help visually to standardize the regression coefficients. You can produce these plots with the following statements.

```
%get_bpls(est1,dsout=bpls);

%plt_bpls(bpls);

   /*********************************************************/
   /  Standardize the PLS regression coefficients           /
   /*********************************************************/

proc standard data=bpls out=bpls mean=0 std=1 vardef=n;
   var b:;

data bpls; set bpls;
   array b b:;
      do i = 1 to dim(b); b{i} = b{i} / 27; end; drop i;
run;
```

```
   /*********************************************************/
   /  Plot the standardized PLS regression coefficients     /
   /*********************************************************/

%plt_bpls(bpls);

   /*********************************************************/
   /  Get VIP and plot it across frequencies                /
   /*********************************************************/

%get_vip(est1,dsvip=vip_data);

%plot_vip(vip_data);
```

The standardized coefficient and VIP plots appear in Figures 19 and 20.

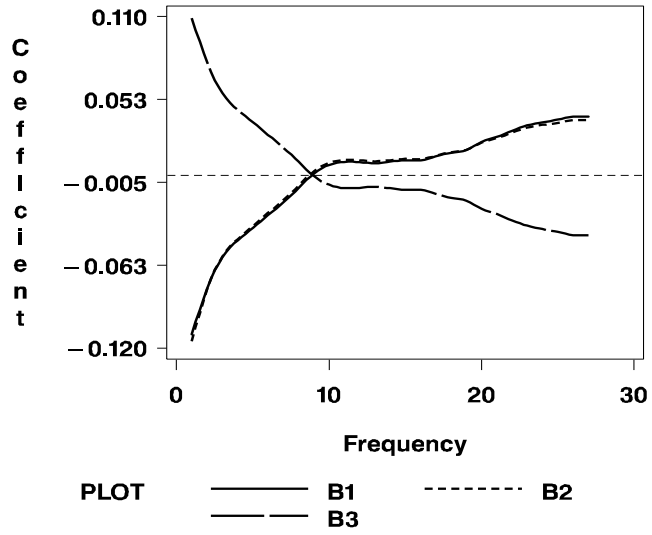

Figure 19: Standardized Regression Coefficient vs. Frequency

When you standardize to take into account location and scale differences in the responses, the resulting coefficient plot (Figure 19) shows very interesting relationships. The predictions for standardized LS and HA are essentially the same linear combination of predictors while the prediction for standardized DT is close to the negative of that linear combination.

The VIP plot shows that all frequencies are important, as the VIP is uniformly larger than 0.8.

Prediction of New Observations

To check the validity of the model, you can use it to predict responses for observations 17 and 18, which were not used in the original model. The following statements make predictions for these observations based on the original data set, calculate the residuals, print them, and plot them versus the predicted values.

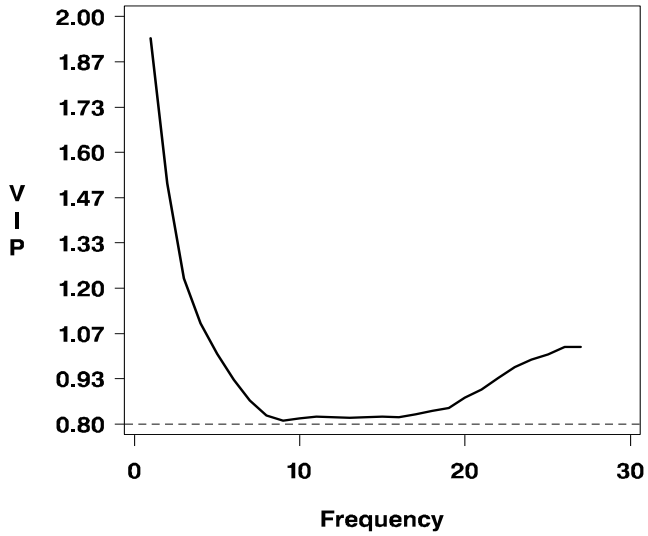

Figure 20: Variable Importance for the Projection for each Frequency

```
   /*********************************************************/
   /  Refit the model with missing values at the points      /
   /  to be predicted.                                       /
   /*********************************************************/

data mswkal_b; set mswkal;
   if n > 16 then do;    *** for predictions ***;
      ls=.;
      ha=.;
      dt=.;
   end;

proc pls data=mswkal_b method=pls outmodel=est2 cv=split(8)
         cvtest(stat=press);
   model &yvars=&xvars;
   output out=outpls2 p=yhat1-yhat3 yresidual=yres1-yres3
          xresidual=xres1-xres27 xscore=xscr yscore=yscr
          stdy=stdy stdx=stdx h=h press=press t2=t2
          xqres=xqres yqres=yqres;
run;

   /*********************************************************/
   /   Put the predicted values and actual observations in   /
   /   the same data set.                                    /
   /*********************************************************/

data outpls2a; set outpls2(keep=yhat1 yhat2 yhat3);
   n=_N_;
run;

data mswkal_c; set mswkal(keep=LS HA DT); n=_N_; run;

data predict; merge mswkal_c outpls2a; by n; run;
```

```
   /*********************************************************/
   /  Calculate the residuals at the points in the test set. /
   /*********************************************************/

data predict; set predict;
   yres1=LS-yhat1;
   yres2=HA-yhat2;
   yres3=DT-yhat3;
run;

proc print data=predict; run;

   /*********************************************************/
   /  Compare the test set and training set residuals.       /
   /*********************************************************/

%res_plot(predict);
```

The residual plots appear in Figures 21, 22, and 23. The printed output is omitted.

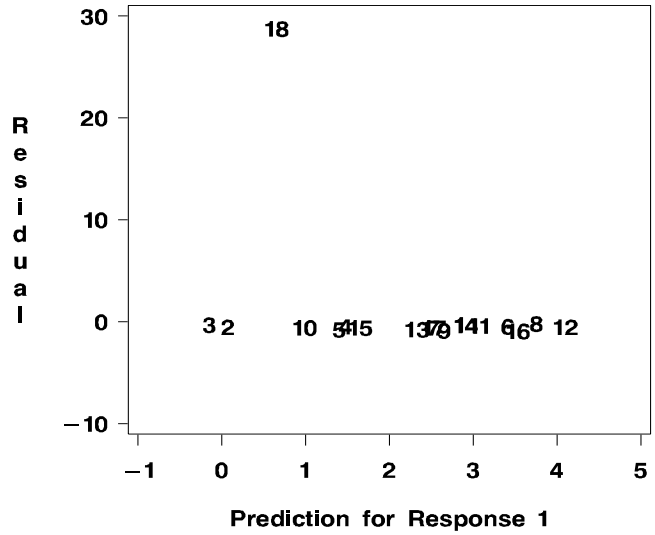

Figure 21: Residuals vs. Predicted Value of LS

You can see from the residual plot that the model predicts observation 17 very well, but it predicts observation 18 very poorly. Observation 18 could be an outlier, or it could be that observation 18 is just far from the other observations in terms of X. Note also that for all observations, modeling of DT is less successful than it is for the other two responses. However, if you add more PLS components, it does not to help model DT significantly better, and it makes the prediction of observation 18 even worse.

To discern why the model doesn't fit observation 18 well, you can calculate the distance between the observation and the model for the predictors. The following statements calculate and plot these distances for each observation.

```
%get_dmod(outpls2,dsdmod=d_mod,qresname=qres,id=n);
```

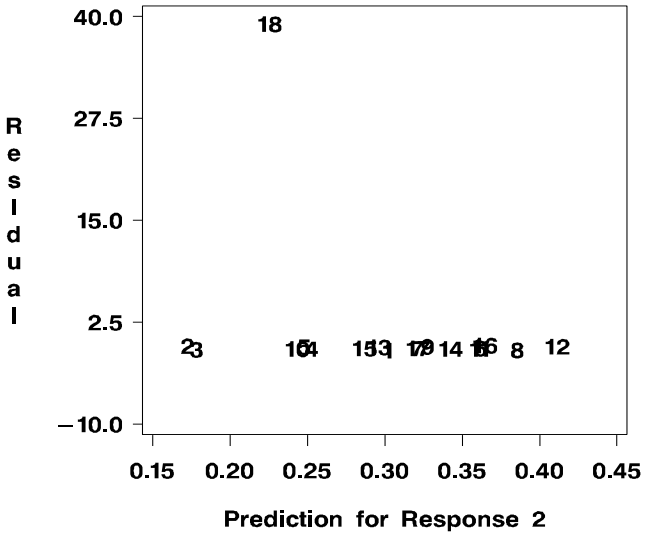

Figure 22: Residuals vs. Predicted Value of HA

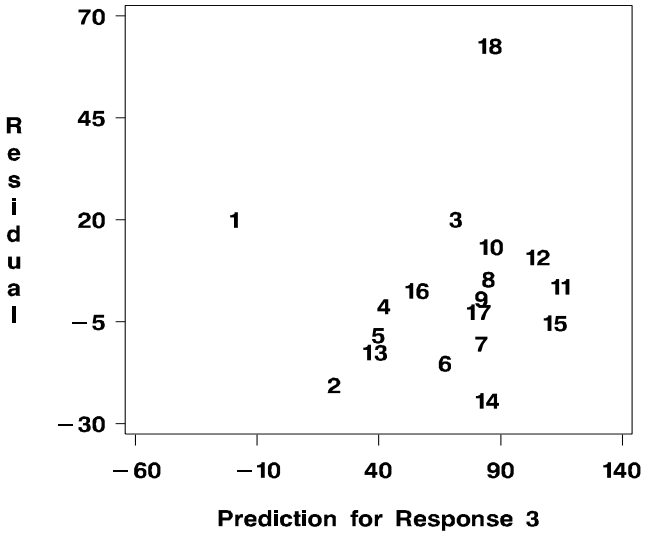

Figure 23: Residuals vs. Predicted Value of DT

```
proc gplot data=d_mod;
   plot dmodx*n;
   symbol1 i=needles v=dot;
run;
```

The plot appears in Figure 24.

Note that observation 18 is three times as far from the model as any other point in the data set. This explains why the model is not appropriate for this observation. Looking at the values of the responses, you can also see that the values for observation 18 are much larger than those of the rest of the data, especially in the case of HA.

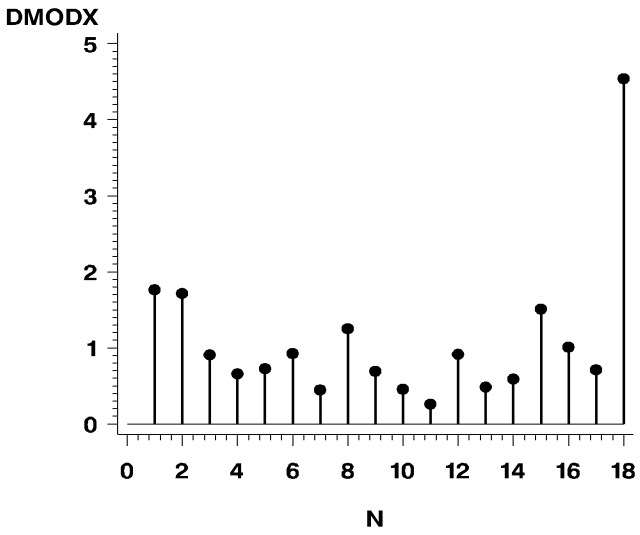

Figure 24: Distances from each Observation to the Model for X

Conclusion

This example demonstrates that partial least squares enables you to calibrate an instrument to estimate concentrations of chemical compounds based on the spectrograph readings that the sample produces. For this example, you can estimate the amounts of LS, HA, and DT based on linear combinations of spectrograph readings at the 27 frequencies, provided the readings are reasonably close to the model for the original 16 observations.

Example 3. Spectrometric Calibration (Lab Data)

Introduction

This example demonstrates additional issues in spectrometric calibration. The data set (Umetrics, 1995) contains spectrographic readings on 33 samples containing known concentrations of two amino acids, tyrosine and tryptophan. The spectra are measured at 30 frequencies across the overall range of frequencies.

Unlike in the previous example, these data were created in a lab. The concentrations were fixed in order to provide a wide range of applicability for the model. The predictors (X) have been logarithmically transformed by log(X + 0.001) and the responses (Y) have also been logarithmically transformed by log(Y) if Y > 0 or set to log(10^{-8}) if Y = 0. The data originally came from McAvoy et al. (1989).

The statements to create a SAS data set named FLUOR5 from these data are supplied in Appendix 1.

First Model Fit

In this example, as in Example 2, you would like to fit a PLS model in order to find linear combinations of the spectra that will serve as predictors for the concentrations of the analytes. Thirteen observations with a total concentration of 3×10^{-5} and five observations with a total concentration of 10^{-4} are used to build the model. To test the validity of the model outside this range of total concentration, predictions are made for seven observations with a total concentration of 10^{-6}, seven with a total concentration of 10^{-5}, and one with a total concentration of 10^{-4}. For each level of total concentration, the levels of tyrosine and tryptophan vary inversely.

As with the other example, a good approach is to use the CVTEST(STAT=PRESS) option with cross-validation groups chosen by CV=SPLIT(9).

The following statements fit the model to the chosen 18 observations, which are observations 15-32 in the FLUOR5 data set. (See Appendix 1.) All macros called in this example appear in Appendix 2.

```
data fluor5a; set fluor5;
   if (15 <= n and n <= 32);
run;

   %let title1=PLS Analysis of Umetrics Fluorescence Example;
   %let xvars=f1 f2 f3 f4 f5 f6 f7 f8 f9 f10 f11 f12 f13 f14
             f15 f16 f17 f18 f19 f20 f21 f22 f23 f24 f25 f26
             f27 f28 f29 f30;
   %let yvars=tot_log tyr_log try_log;
   %let ypred=yhat1 yhat2 yhat3;
   %let yres=yres1 yres2 yres3;
   %let predname=yhat;
```

```
   %let resname=res;
   %let xscrname=xscr;
   %let yscrname=yscr;
   %let num_y=3;
   %let num_x=30;

   /*************************************************/
   /  Fit the PLS model to observations 15-32       /
   /*************************************************/

proc pls data=fluor5a method=pls outmodel=est1 cv=split(9)
   cvtest(stat=PRESS);
   model tot_log tyr_log try_log = f1-f30;
   output out=outpls p=yhat1-yhat3 yresidual=yres1-yres3
          xresidual=xres1-xres30 xscore=xscr yscore=yscr
          stdy=stdy stdx=stdx h=h press=press t2=t2
          xqres=xqres yqres=yqres;
run;

   %let lv=6;       *** Number of PLS components in model ***;
```

The PLS output appears in Output 3.1 and Output 3.2.

Output 3.1. Cross-Validation Results

1

The PLS Procedure
Cross Validation for the Number of Latent Variables

Number of Latent Variables	Root Mean PRESS	Test for larger residuals than minimum Prob > PRESS
0	1.0809	0
1	0.8270	0
2	0.5366	0
3	0.4110	0
4	0.3266	0.00100
5	0.2870	0.00200
6	0.2627	1.0000
7	0.2765	0.2030
8	0.3284	0.0660
9	0.3621	0.0260
10	0.3859	0.0150
11	0.3991	0.0400
12	0.4446	0.0110
13	0.5215	0.00200
14	0.5724	0
15	0.6013	0
16	0.6013	0
17	0.6013	0
18	0.6013	0
19	0.6013	0
20	0.6013	0
21	0.6013	0
22	0.6013	0
23	0.6013	0
24	0.6013	0
25	0.6013	0
26	0.6013	0
27	0.6013	0
28	0.6013	0
29	0.6013	0
30	0.6013	0

Minimum Root Mean PRESS = 0.26268 for 6 latent variables
Smallest model with p-value > 0.1: 6 latent variables

Output 3.2. Percentages of Variation Explained by Model

The PLS Procedure
Percent Variation Accounted For

Number of Latent Variables	Model Effects Current	Model Effects Total	Dependent Variables Current	Dependent Variables Total
1	81.1648	81.1648	48.3383	48.3383
2	16.8119	97.9768	32.5471	80.8854
3	1.7639	99.7406	11.4465	92.3320
4	0.1951	99.9357	3.8334	96.1654
5	0.0276	99.9633	1.6857	97.8510
6	0.0132	99.9765	0.7245	98.5755

You can see from the output that PROC PLS selected a model with six PLS components (latent variables) that explain nearly all of the variation in both predictors and responses. Actually, the first three components capture most of the variation, so it would be good to keep this in mind when doing the analysis.

To check for possible improvements in the model, you can use the following statements to examine plots of Y-scores versus the corresponding X-scores.

```
%plot_scr(outpls);
```

The plots for the first three PLS components appear in Figures 25, 26 and 27. Recall that the numbers on the plot represent observation numbers in the data set.

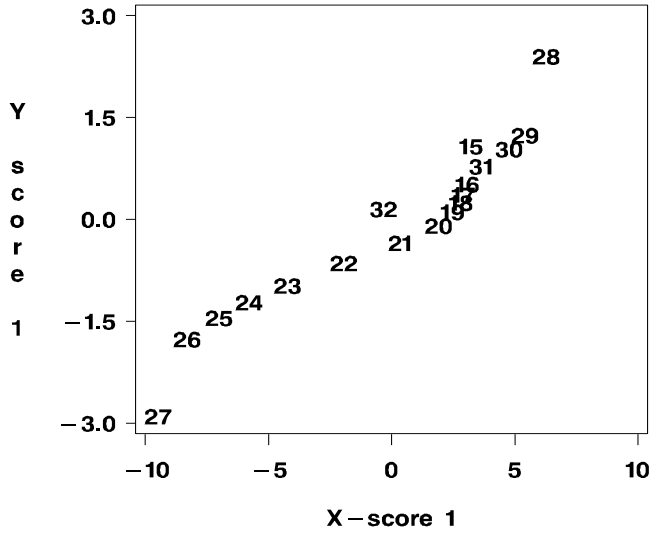

Figure 25: First X- and Y-scores for Fluorescence Model

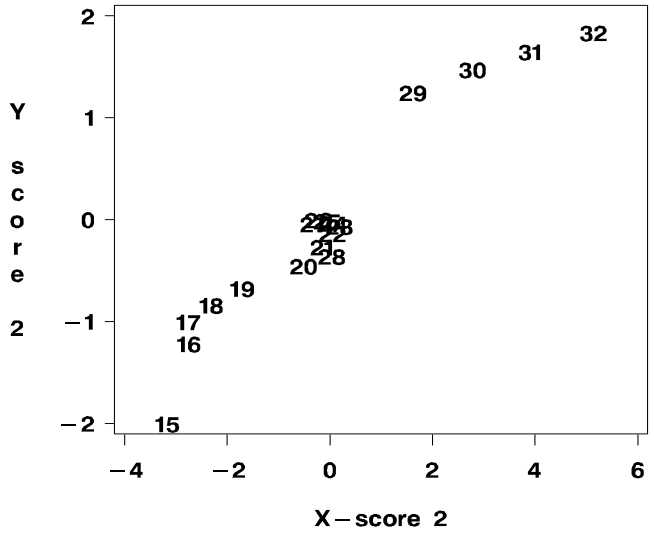

Figure 26: Second X- and Y-scores for Fluorescence Model

In Figures 25-27, notice the interesting patterns formed by the scores. Recall that observations 15-27 all have the same total concentration and observations 28-32

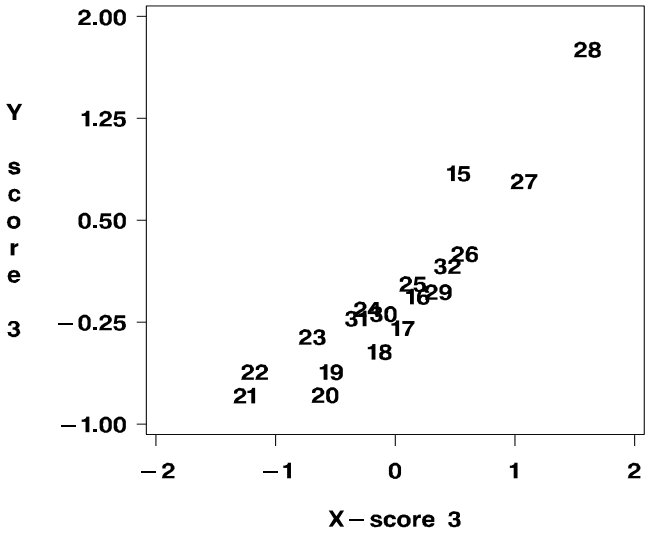

Figure 27: Third X- and Y-scores for Fluorescence Model

also have the same total concentration. Each group forms a distinctive pattern due to the fact that within each group tyrosine gradually increases while the tryptophan concentration gradually decreases from one observation to the next.

You can see from the score plots that the first three components have considerably higher correlated X- and Y-scores, as the R-square table suggested earlier. The score plot for the third component hints at curvature. You can test for curvature by taking the third X- and Y-scores from the PLS OUTPUT data set and fitting a regression of the Y-score on the X-score with a quadratic term in X.

```
proc glm data=outpls;
   model yscr3=xscr3 xscr3*xscr3;
run;
```

The output from the GLM procedure (not shown) reveals that there is a statistically significant quadratic relationship, but incorporating this into the model changes very little; thus, quadratic terms in the frequencies are not added to the model.

You can plot as many pairs of consecutive X-scores against each other as you would like by calling the **%plotxscr** macro and specifying the **MAX_LV** parameter to be the last PLS component to be included in a plot. For example, if **MAX_LV**=3, the macro generates plots for X-score 2 versus X-score 1 and X-score 3 versus X-score 2.

```
%plotxscr(outpls,max_lv=3);
```

The plots are shown in Figures 28 and 29. The pattern between X-scores 1 and 2 again shows the two groups based on the total concentration and the pattern due to the increasing proportion of tyrosine (TYR) in the mix. You might consider analyzing the two groups separately, but this would further limit the applicability of the model to differing amounts of total concentration.

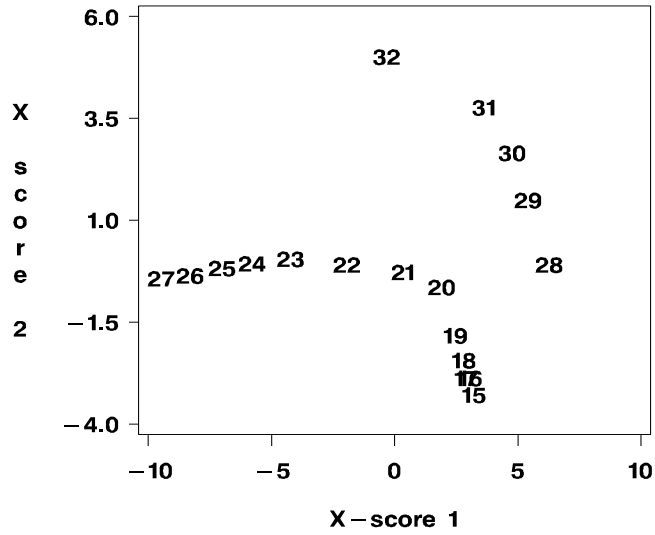

Figure 28: First and Second X-scores for Fluorescence Model

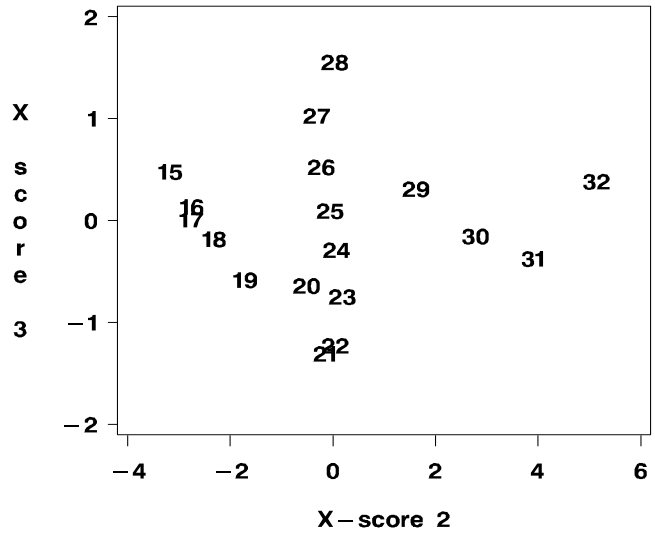

Figure 29: Second and Third X-scores for Fluorescence Model

X-scores 2 and 3 form an interesting pattern in Figure 29, but observation 20 appears to deviate from it. This indicates it might be worthwhile to check observation 20 for accuracy. To study the source of the patterns in the score plots, you can plot the residuals versus the predicted values and a normal quantile plot of the residuals using the following macro calls.

```
%res_plot(outpls);
%nor_plot(outpls);
```

The three plots of the residuals versus predicted values appear in Figures 30, 31, and 32, while the three normal quantile plots appear in Figures 33, 34, and 35.

The plot of residuals versus predicted values for the first response (**TOT_LOG**) looks granular, but this happens because there are only two values for **TOT_LOG**.

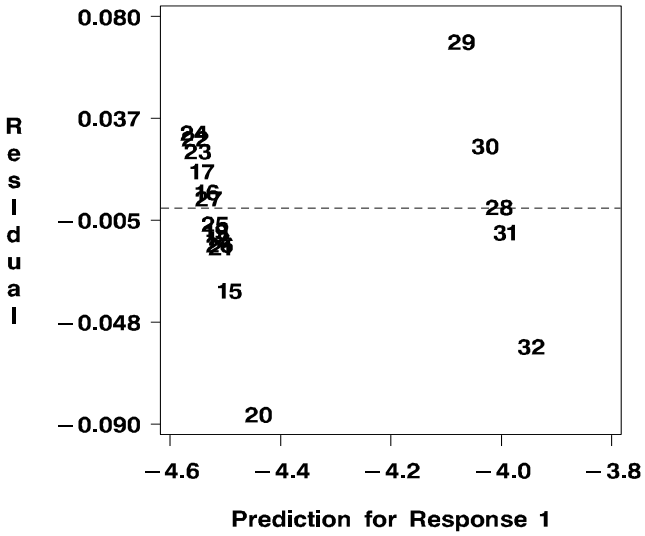

Figure 30: Residuals vs. Predicted Value of **TOT_LOG**

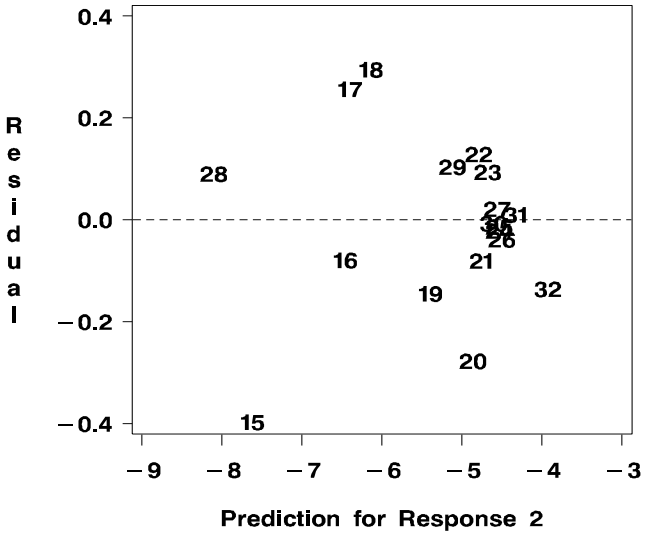

Figure 31: Residuals vs. Predicted Value of **TYR_LOG**

The residual versus predicted value plots for **TYR_LOG** and **TRY_LOG** in Figures 31 and 32 show that the residuals may be heteroscedastic. In this case, it appears that there is less variability in **TYR_LOG** and **TRY_LOG** for higher relative concentrations of TYR and TRY, respectively. Also, the variability seems to decrease when the total concentration increases.

In the normal quantile plot for **TOT_LOG**, observations 15, 20, 29 and 32 do not fit the pattern of the rest of the observations. Observations 15 and 20 do not fit well in the normal plot for the **TYR_LOG** residuals either. In observation 15, the amino acid is pure tryptophan, so it is not surprising that the residual for tyrosine is nonnormal. The normal plot for the **TRY_LOG** residuals looks fine.

Since the normal plots indicate possible outliers for several observations, it might be useful to look at the distance of each observation from the model. The following statements produce the appropriate plots.

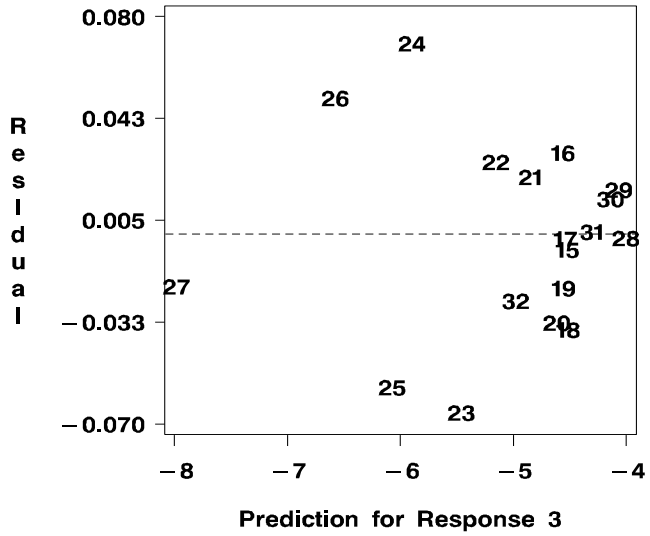

Figure 32: Residuals vs. Predicted Value of **TRY_LOG**

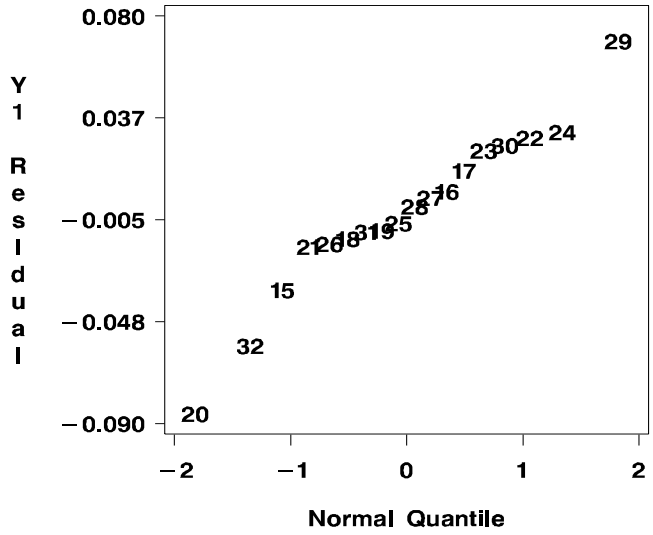

Figure 33: Normal Quantile Plot of **TOT_LOG** Residuals

```
%get_dmod(outpls,dsdmod=d_mod,qresname=qres,id=n);

proc gplot data=d_mod;
   plot dmodx*n;
   symbol1 i=needles v=dot;
run;

proc gplot data=d_mod;
   plot dmody*n;
   symbol1 i=needles v=dot;
run;
```

The plots appear in Figures 36 and 37. In the figures, no observation stands out from the others in terms of distance from the model in either X or Y.

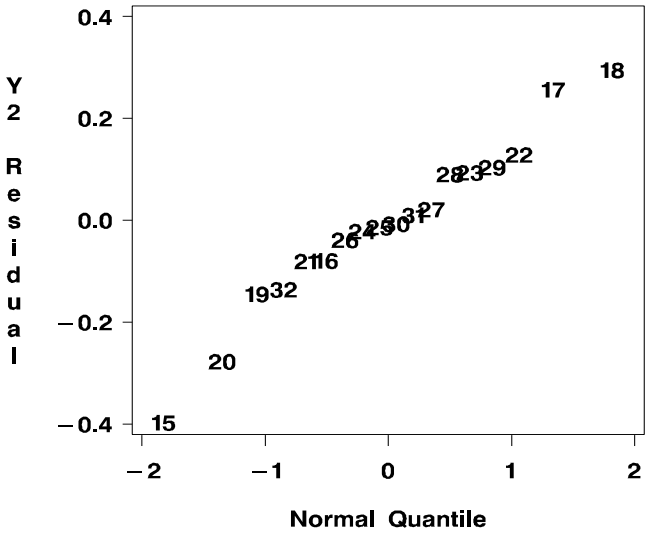

Figure 34: Normal Quantile Plot of **TYR_LOG** Residuals

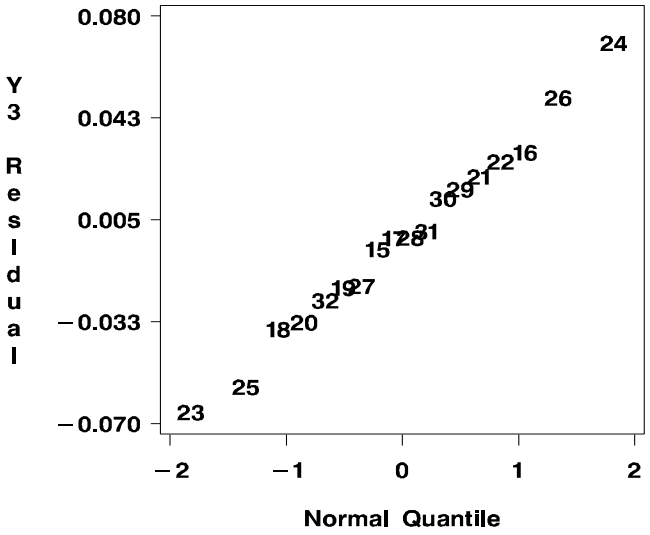

Figure 35: Normal Quantile Plot of **TRY_LOG** Residuals

When the score plots reveal irregularities, the loadings plots are especially useful for diagnosing problems. First, you can plot the weights and loadings for the predictors. Because the predictors are ordered by frequency, it makes sense to plot the weights and loadings versus frequency for each PLS component. You can do this using the following statements.

```
/*********************************************************/
/  Compute the X-Weights for each PLS component          /
/*********************************************************/

%get_wts(est1,dsxwts=xwts);
```

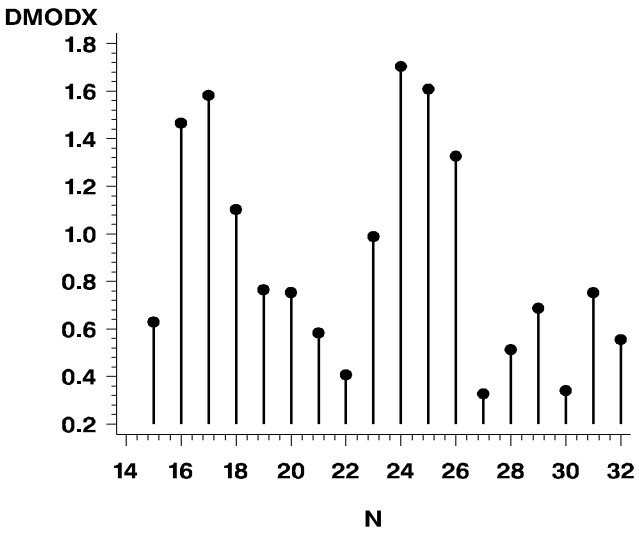

Figure 36: Distances from each Observation to the Model for X

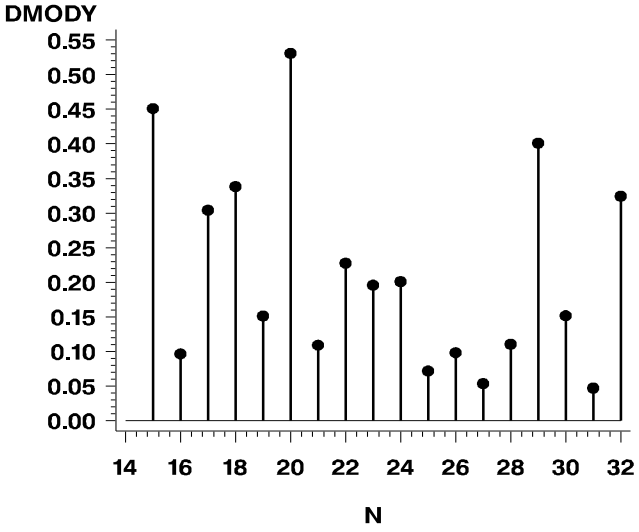

Figure 37: Distances from each Observation to the Model for Y

```
   /*********************************************************/
   /  Plot the X-weights vs. the frequency on the same axes  /
   /*********************************************************/

%pltwtfrq(xwts,plotyvar=w,plotxvar=n,max_lv=&lv,
          label=Weight);

   /*********************************************************/
   /  Compute X-loadings p1-p6 for the six components        /
   /*********************************************************/

%getxload(est1,dsxload=xloads);
```

```
/*********************************************************/
/  Plot the X-loadings for each component vs. frequency   /
/*********************************************************/

%pltwtfrq(xloads,plotyvar=p,plotxvar=n,max_lv=&lv,
          label=Loading);
```

The plot of the X-loadings versus the frequency appears in Figure 38. The X-weights plot is very similar.

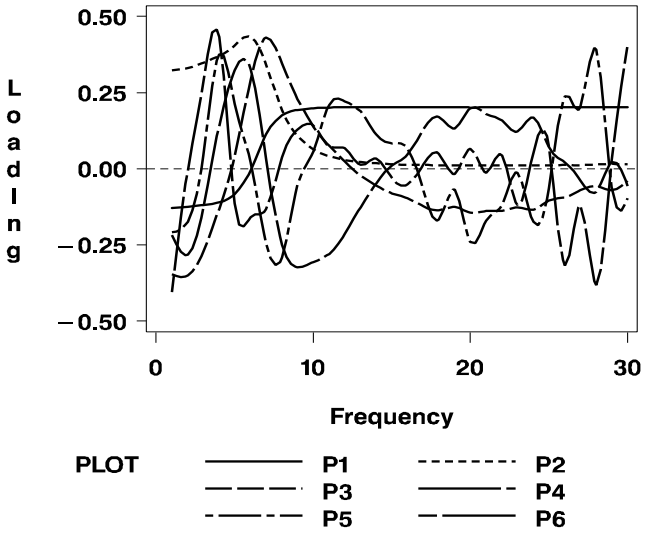

Figure 38: X-Loadings Across Frequencies for Fluorescence Model

The loadings plot shows that the PLS model gives somewhat larger importance to the lower frequencies. However, it does give nonzero weight to all frequencies.

Note from the figure that the loading curves are much bumpier for components 4-6 than for components 1-3. This raises the possibility that components 4-6 are just modeling noise. Recall that the R-square table showed much smaller improvements to the fit for components 4-6.

This plot may seem somewhat cluttered, especially in black and white. If you want to see the plot of loadings for only the first three PLS factors, you can reinvoke the **%pltwtfrq** macro with **MAX_LV=3**.

The X-loadings plot appears to indicate that the lower frequencies are the most important for the model. To further study how the frequencies contribute to the model, you can plot the PLS coefficients and the VIP using the following statements.

```
%get_bpls(est1,dsout=bpls);
```

```
   /*********************************************************/
   /  Standardize the PLS regression coefficients           /
   /*********************************************************/

proc standard data=bpls out=bpls mean=0 std=1 vardef=n;
   var b:;

data bpls; set bpls;
   array b b:;
      do i = 1 to dim(b); b{i} = b{i} / 27; end; drop i;
run;

   /*********************************************************
   /  Plot the standardized PLS regression coefficients     /
   *********************************************************/

%plt_bpls(bpls);

   /*********************************************************
   /  Get VIP and plot it across frequencies                /
   *********************************************************/

%get_vip(est1,dsvip=vip_data);

%plot_vip(vip_data);
```

Figures 39 and 40 display the coefficient and VIP plots, respectively.

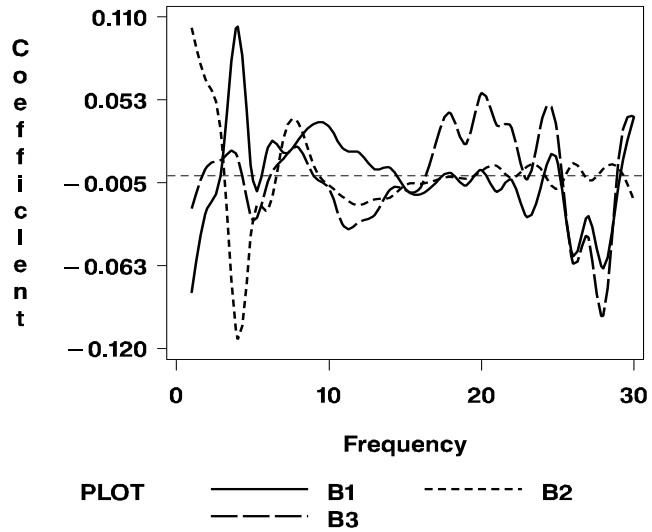

Figure 39: Standardized Regression Coefficient vs. Frequency

Figures 39 and 40 show that the first ten frequencies have the most impact on the model, while the highest frequencies have slightly more impact than the middle frequencies. The coefficients for each of the three responses form a fairly bumpy curve, indicating again that partial least squares regression may be attempting to

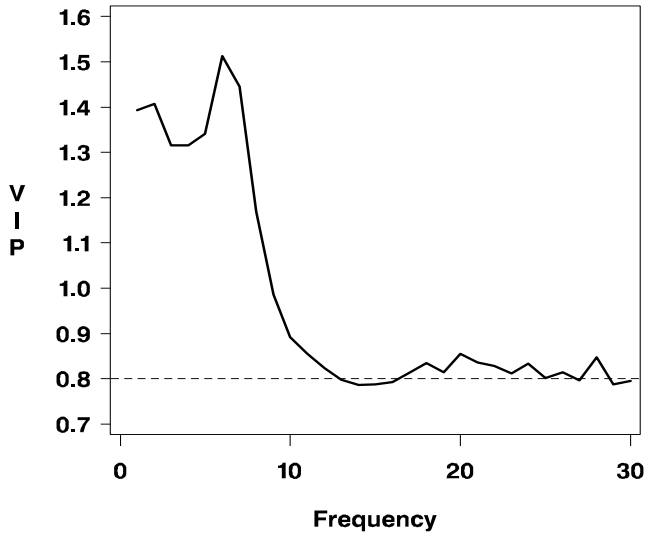

Figure 40: Variable Importance for the Projection for each Frequency

model noise.

The R-square table, the X-weights plot, and the PLS coefficients plot have all given evidence that the model is overfit, which means it fits the observations used in modeling well but will predict new observations poorly. To check this, you can use the model to predict observations 1-14 and 33.

Prediction of New Observations

The following statements set the responses in observations 1-14 and 33 to missing, then fits the same PLS model to observations 15-32 and makes predictions for observations 1-14 and 33. It then compares the predictions for the new observations to their actual values and plots the residuals versus the predicted values.

```
   /*********************************************************/
   /  Refit the model with missing values at the points      /
   /  to be predicted.                                       /
   /*********************************************************/

data fluor5b; set fluor5;
   if ( n <= 14 or n = 33) then do;    *** for predictions;
      tot_log=.;
      tyr_log=.;
      try_log=.;
   end;

proc pls data=fluor5b method=pls outmodel=est2 cv=split(9)
         cvtest(stat=PRESS);
   model tot_log tyr_log try_log=f1-f30;
   output out=outpls2 p=yhat1-yhat3 yresidual=yres1-yres3
          xresidual=xres1-xres30 xscore=xscr yscore=yscr
          stdy=stdy stdx=stdx h=h press=press t2=t2
          xqres=xqres yqres=yqres;
```

```
run;

   /*********************************************************/
   /   Put the predicted values and actual observations in   /
   /   the same data set.                                    /
   /*********************************************************/

data outpls2a; set outpls2(keep=yhat1 yhat2 yhat3);
   n=_N_;
run;

data fluor5c; set fluor5(keep=TOT_LOG TYR_LOG TRY_LOG);
   n=_N_;
run;

data predict; merge fluor5c outpls2a; by n; run;

   /*********************************************************/
   /  Calculate the residuals at the points in the test set. /
   /*********************************************************/

data predict; set predict;
   yres1=TOT_LOG-yhat1;
   yres2=TYR_LOG-yhat2;
   yres3=TRY_LOG-yhat3;
run;

   /*********************************************************/
   /  Compare the test set and training set residuals.       /
   /*********************************************************/

%res_plot(predict);
```

The residual plots for the three responses for all observations based on the model for observations 15-32 appear in Figures 41, 42, and 43.

The residual plots for the second and third responses, **TYR_LOG** and **TRY_LOG**, show much more variability in predicting the new observations than in predicting those observations used in modeling. This indicates that the model for observations 15-32 may not apply to observations 1-14 and 33. The distances of the new observations to the PLS model for the predictors illuminate this further, seen in the plots produced by the following statements.

```
%get_dmod(outpls2,dsdmod=d_mod,qresname=qres,id=n);

proc gplot data=d_mod;
   plot dmodx*n;
   symbol1 i=needles v=dot;
run;
```

The plot appears in Figure 44.

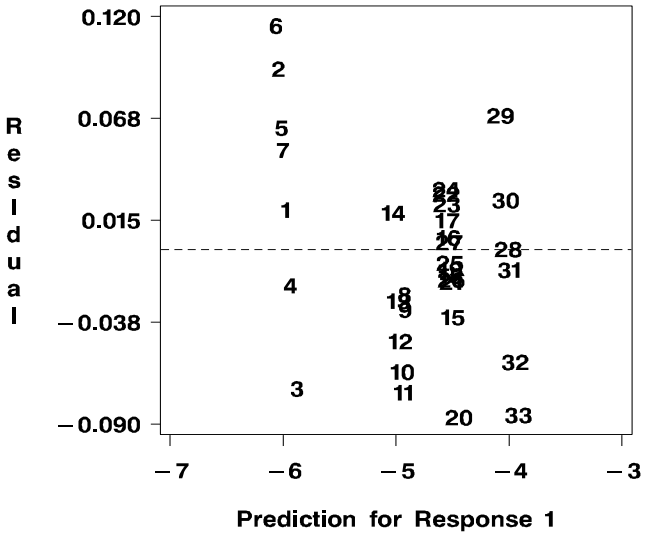

Figure 41: Residual Plot of **TOT_LOG** for all Observations

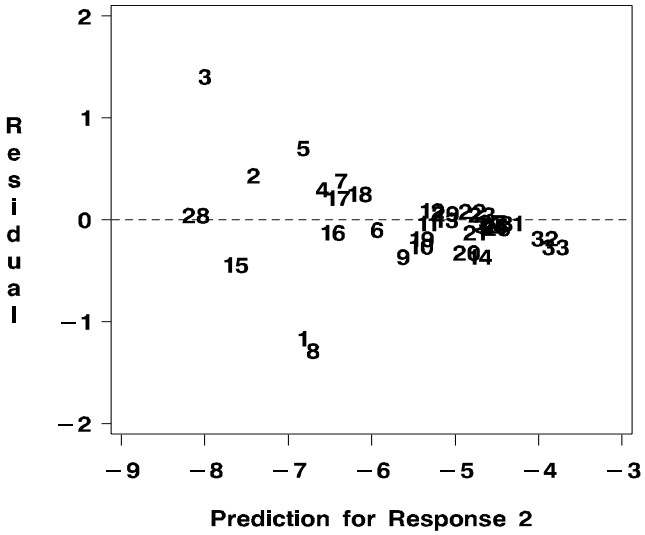

Figure 42: Residual Plot of **TYR_LOG** for all Observations

The plot shows that the new observations are much farther from the model than the first set of observations. So now the question is, how can you improve this model? Recall that the improvement in the R-square for the responses tailed off considerably after the third PLS component, even though cross-validation recommended a six-term model. Also recall the evidence from the weights and loadings plots, as well as the regression coefficients, which indicated that components 4-6 may be modeling noise. Thus, a natural approach would be to fit a PLS model with three components.

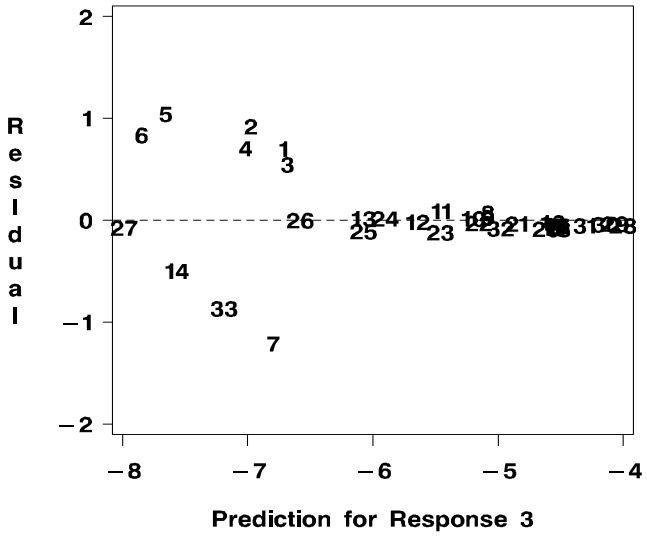

Figure 43: Residual Plot of **TRY_LOG** for all Observations

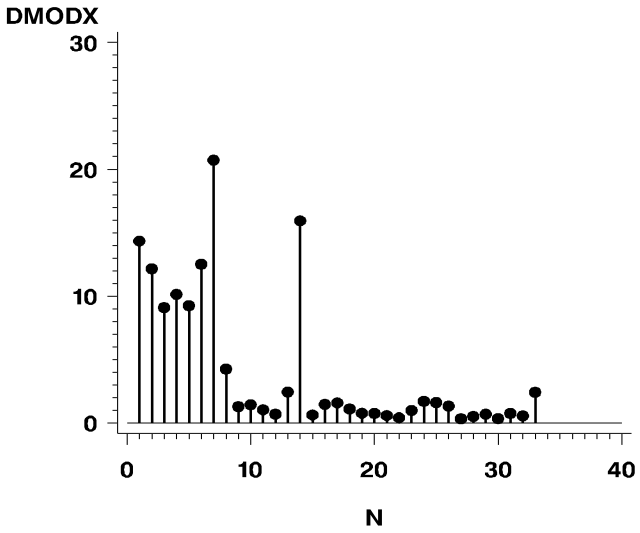

Figure 44: Distances to the Model for X Based on Observations 15-32

Second PLS Model

You can fit a three-term PLS model by specifying **LV**=3.

```
proc pls data=fluor5a method=pls outmodel=est3 lv=3;
   model tot_log tyr_log try_log = f1-f30;
   output out=outpls3 p=yhat1-yhat3 yresidual=yres1-yres3
          xresidual=xres1-xres30 xscore=xscr yscore=yscr stdy=stdy
          stdx=stdx h=h press=press t2=t2 xqres=xqres yqres=yqres;
run;

   %let lv=3;        *** Number of PLS components ***;
```

The output is the same as the first three components in Output 3.2

You can study the X-loadings for each PLS component across frequencies by again plotting the loadings with the following commands.

```
%getxload(est3,dsxload=xloads3);

%pltwtfrq(xloads3,plotyvar=p,plotxvar=n,max_lv=&lv,
          label=Loading);
```

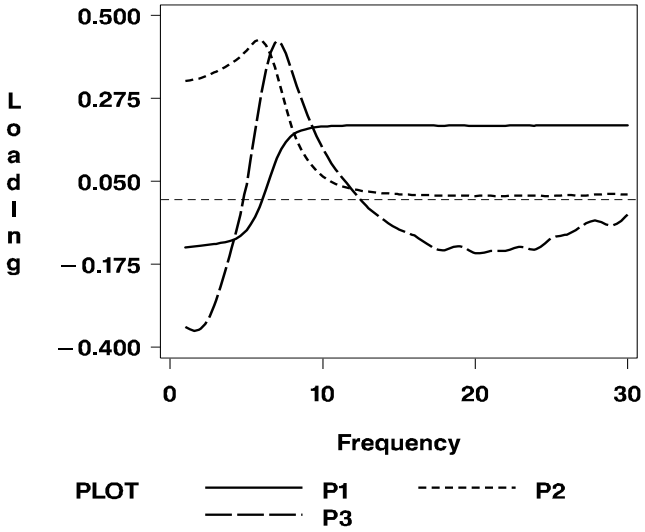

Figure 45: X-Loadings Across Frequencies for Fluorescence Model 2

The loadings plot appears in Figure 45. Note that the first PLS component appears to contrast frequencies 1-7 with the remaining ones. The second component appears to represent the weighted average of the first 10 frequencies. The third component appears to be a contrast between frequencies 1-5 and 6-10 or so.

You can judge the impact of simplifying the model on the PLS coefficients by looking at the coefficient plot. The following statements plot these coefficients and also plot the Variable Importance for the Projection (VIP) for the new model.

```
   /*********************************************************/
   / Get B(PLS), the matrix of regression coefficients      /
   /*********************************************************/

%get_bpls(est3,dsout=bpls3);

   /*********************************************************/
   /  Standardize the PLS regression coefficients           /
   /*********************************************************/

proc standard data=bpls3 out=bpls3 mean=0 std=1 vardef=n;
   var b:;

data bpls3; set bpls3;
   array b b:;
      do i = 1 to dim(b); b{i} = b{i} / 30; end; drop i;
```

```
run;

   /*********************************************************/
   /  Plot the standardized PLS regression coefficients     /
   /*********************************************************/

%plt_bpls(bpls3);

   /*********************************************************/
   /  Get VIP and plot it across frequencies                /
   /*********************************************************/

%get_vip(est3,dsvip=vipdata3);

%plot_vip(vipdata3);
```

The coefficient and VIP plots appear in Figures 46 and 47, respectively.

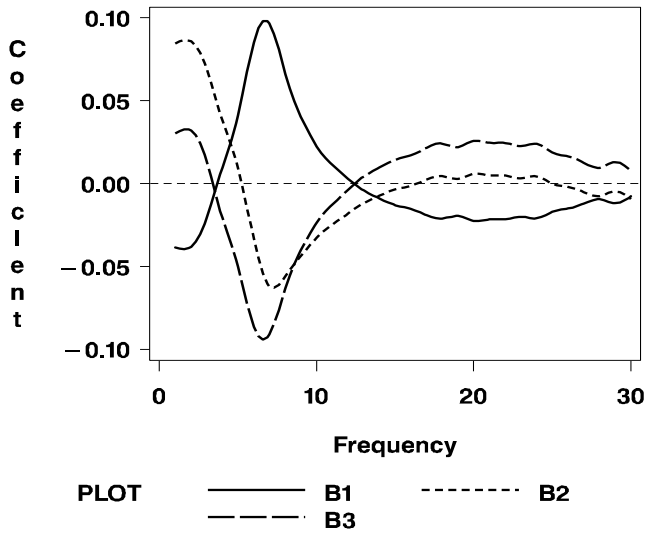

Figure 46: Standardized Regression Coefficients for Second Model

Notice the dramatic difference in the coefficient plot by comparing it to the one generated in the six-component model. It is much smoother than before. The VIP plot shows that the new model still emphasizes the lower frequencies but uses all frequencies, since VIP is larger than the 0.8 cutoff of Wold (1994).

To see how well this three-term model predicts new observations, you can redo the predictions and plot the residuals.

```
   /*********************************************************/
   /  Redo predictions again                                /
   /*********************************************************/

proc pls data=fluor5b method=pls outmodel=est4 lv=3 noprint;
   model tot_log tyr_log try_log=f1-f30;
   output out=outpls4 p=yhat1-yhat3 yresidual=yres1-yres3
          xresidual=xres1-xres30 xscore=xscr yscore=yscr
```

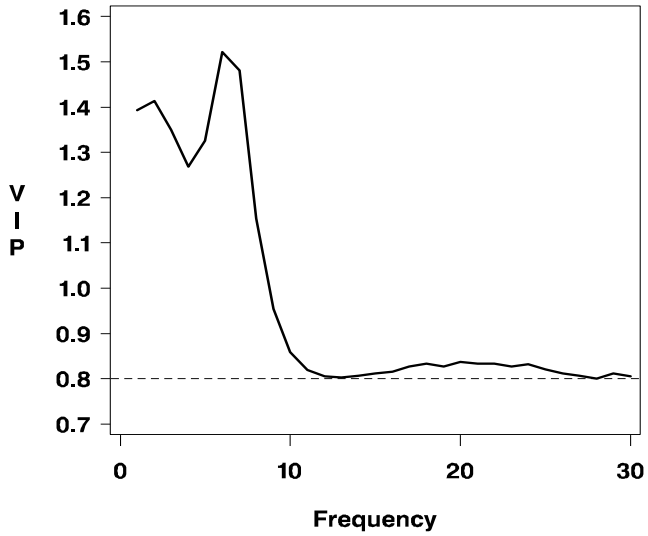

Figure 47: VIP for Second PLS Model

```
          stdy=stdy stdx=stdx h=h press=press t2=t2
          xqres=xqres yqres=yqres;
run;

   %let title3=Predicting Obs. 1-33 from Fit for Obs. 15-32;

data outpls4a; set outpls4(keep=yhat1 yhat2 yhat3);
   n=_N_;
run;

data fluor5c; set fluor5(keep=TOT_LOG TYR_LOG TRY_LOG);
   n=_N_;
run;

data predict2; merge fluor5c outpls4a; by n; run;

data predict2; set predict2;
   yres1=TOT_LOG-yhat1;
   yres2=TYR_LOG-yhat2;
   yres3=TRY_LOG-yhat3;
run;

%res_plot(predict2);
```

The three residual plots for all observations based on the three-factor PLS model for observations 15-32 appear in Figures 48, 49, and 50.

The residual plot for **TOT_LOG** looks about the same, but those for **TYR_LOG** and **TRY_LOG** show that this model does a much better job of predicting observations 1-14 and 33, which include observations with smaller total concentration than those in the set used for the model fit.

On the **TYR_LOG** (response 2) residual plot, observations 1 and 8 are more outlying than the rest. However, it is interesting that these are the observations that contain no

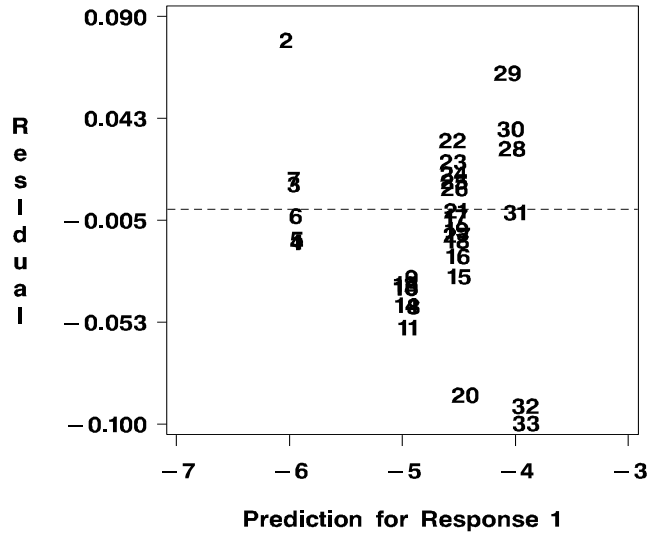

Figure 48: Second Residual Plot of **TOT_LOG** for all Observations

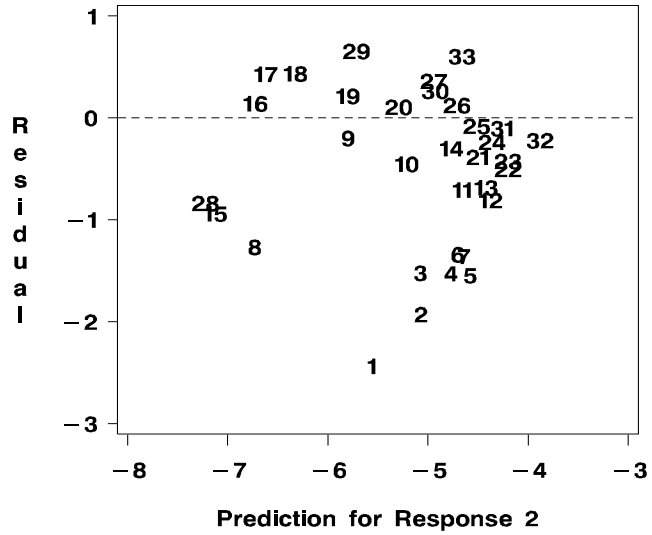

Figure 49: Second Residual Plot of **TYR_LOG** for all Observations

tyrosine, the quantity you are trying to predict. The same problem shows up on the plot for **TRY_LOG** (response 3). Here, new observations 7, 14, and 33 have outlying residuals, but note that they are the observations with no tryptophan. Observation 27 comes from the set used for the model, but it also contains no tryptophan.

Conclusion

This example demonstrates that although cross-validation helps in selecting the number of PLS components, you should not use it blindly. The model recommended by the cross-validation test overfit the data and failed to predict new observations well. However, the R-square table, the X-weight plot, and the coefficient plot all were useful in diagnosing overfitting as a possible problem.

The model with three PLS components predicted new observations well, even though

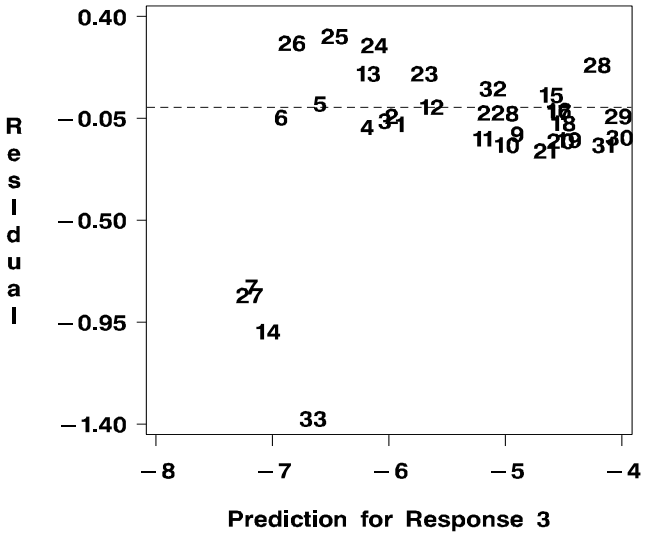

Figure 50: Second Residual Plot of **TRY_LOG** for all Observations

the new samples (except observation 33) had much lower total concentration. The only exceptions were the cases where there was no tyrosine or no tryptophan. In those cases it predicted a nonzero quantity for the given amino acid. Overall, the three-term model does very well.

References

Lindberg, W., Persson, J-A., and Wold, S. (1983), ‘‘Partial Least-Squares Method for Spectrofluorimetric Analysis of Mixtures of Humic Acid and Ligninsulfonate’’ *Analytical Chemistry* 55, 643–648.

McAvoy, T. J., Wang, N. S., Naidu, S., Bhat, N., Gunter, J., and Simmons, M. (1989), ‘‘Interpreting Biosensor Data via Backpropagation,’’ *International Joint Conference on Neural Networks*, 1, 227–233.

Shao, J. (1993), ‘‘Linear Model Selection by Cross-Validation,’’ *Journal of the American Statistical Association*, 88, 486–494.

Tobias, R. (1995), ‘‘An Introduction to Partial Least Squares Regression,’’ in *Proceedings of the Twentieth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc., 1250–1257.

Ufkes, J. G. R., Visser, B. J., Heuver, G., and Van Der Meer, C. (1978), ‘‘Structure-Activity Relationships of Bradykinin-Potentiating Peptides,’’ *European Journal of Pharmacology*, 50, 119.

Ufkes, J. G. R., Visser, B. J., Heuver, G., Wynne, H. J., and Van Der Meer, C. (1982), ‘‘Further Studies on the Structure-Activity Relationships of Bradykinin-Potentiating Peptides,’’ *European Journal of Pharmacology*, 79, 155.

Umetrics, Inc. (1995), *Multivariate Analysis (3-day course)*, Winchester, MA.

Wold, S. (1994), ‘‘PLS for Multivariate Linear Modeling,’’ *QSAR: Chemometric Metods in Molecular Design. Methods and Principles in Medicinal Chemistry. (Ed. H. van de Waterbeemd)*, Weinheim, Germany: Verlag- Chemie.

Appendix 1: Data Sets

For each of the three data sets, the variable **n** represents the observation number and appears at the extreme left.

```
data penta;
   input n obsnam $ S1 L1 P1 S2 L2 P2 S3 L3 P3 S4 L4 P4
         S5 L5 P5 log_RAI @;
   datalines;
1  VESSK    -2.6931   -2.5271   -1.2871   3.0777   0.3891   -0.0701
   1.9607   -1.6324   0.5746   1.9607   -1.6324   0.5746   2.8369
                      1.4092    -3.1398       0
2  VESAK    -2.6931   -2.5271   -1.2871   3.0777   0.3891   -0.0701
   1.9607   -1.6324   0.5746   0.0744   -1.7333   0.0902   2.8369
                      1.4092    -3.1398      0.28
3  VEASK    -2.6931   -2.5271   -1.2871   3.0777   0.3891   -0.0701
   0.0744   -1.7333   0.0902   1.9607   -1.6324   0.5746   2.8369
                      1.4092    -3.1398      0.2
4  VEAAK    -2.6931   -2.5271   -1.2871   3.0777   0.3891   -0.0701
   0.0744   -1.7333    0.0902    0.0744  -1.7333   0.0902    2.8369
                      1.4092    -3.1398      0.51
5  VKAAK    -2.6931   -2.5271   -1.2871   2.8369   1.4092   -3.1398
   0.0744   -1.7333    0.0902   0.0744   -1.7333   0.0902    2.8369
                      1.4092    -3.1398      0.11
6  VEWAK    -2.6931   -2.5271   -1.2871   3.0777   0.3891   -0.0701
   -4.7548   3.6521    0.8524   0.0744   -1.7333   0.0902    2.8369
                      1.4092    -3.1398      2.73
7  VEAAP    -2.6931   -2.5271   -1.2871   3.0777   0.3891   -0.0701
   0.0744   -1.7333   0.0902   0.0744   -1.7333    0.0902   -1.2201
                 0.8829    2.2253      0.18
8  VEHAK    -2.6931   -2.5271   -1.2871   3.0777   0.3891   -0.0701
   2.4064    1.7438    1.1057    0.0744   -1.7333   0.0902   2.8369
                1.4092    -3.1398      1.53
9  VAAAK    -2.6931   -2.5271   -1.2871   0.0744   -1.7333   0.0902
   0.0744   -1.7333    0.0902    0.0744   -1.7333   0.0902   2.8369
                     1.4092    -3.1398      -0.1
10 GEAAK    2.2261   -5.3648   0.3049   3.0777   0.3891   -0.0701
   0.0744   -1.7333   0.0902   0.0744   -1.7333   0.0902   2.8369
                 1.4092    -3.1398     -0.52
11 LEAAK    -4.1921   -1.0285   -0.9801   3.0777   0.3891   -0.0701
   0.0744   -1.7333   0.0902   0.0744   -1.7333   0.0902   2.8369
                 1.4092    -3.1398      0.4
12 FEAAK    -4.9217   1.2977   0.4473   3.0777   0.3891   -0.0701
   0.0744   -1.7333   0.0902   0.0744   -1.7333   0.0902   2.8369
                 1.4092    -3.1398      0.3
13 VEGGK    -2.6931   -2.5271   -1.2871   3.0777   0.3891   -0.0701
   2.2261   -5.3648   0.3049   2.2261   -5.3648   0.3049   2.8369
                 1.4092    -3.1398       -1
14 VEFAK    -2.6931   -2.5271   -1.2871   3.0777   0.3891   -0.0701
   -4.9217   1.2977   0.4473   0.0744   -1.7333   0.0902   2.8369
                 1.4092    -3.1398      1.57
15 VELAK     -2.6931   -2.5271   -1.2871   3.0777   0.3891   -0.0701
```

```
   -4.1921   -1.0285   -0.9801   0.0744   -1.7333   0.0902   2.8369
                  1.4092    -3.1398      0.59
16 AAAAA     0.0744   -1.7333   0.0902   0.0744   -1.7333   0.0902
   0.0744   -1.7333   0.0902   0.0744   -1.7333   0.0902   0.0744
                  -1.7333    0.0902      -0.1
17 AAYAA     0.0744   -1.7333   0.0902   0.0744   -1.7333   0.0902
   -1.3944   2.323   0.0139   0.0744   -1.7333   0.0902   0.0744
                  -1.7333    0.0902      0.46
18 AAWAA     0.0744   -1.7333   0.0902   0.0744   -1.7333   0.0902
   -4.7548   3.6521   0.8524   0.0744   -1.7333   0.0902   0.0744
                  -1.7333    0.0902      0.75
19 VAWAA    -2.6931   -2.5271   -1.2871   0.0744   -1.7333   0.0902
   -4.7548   3.6521   0.8524   0.0744   -1.7333   0.0902   0.0744
                  -1.7333    0.0902      1.43
20 VAWAK    -2.6931   -2.5271   -1.2871   0.0744   -1.7333   0.0902
   -4.7548   3.6521   0.8524   0.0744   -1.7333   0.0902   2.8369
                  1.4092    -3.1398      1.45
21 VKWAA    -2.6931   -2.5271   -1.2871   2.8369   1.4092   -3.1398
   -4.7548   3.6521   0.8524   0.0744   -1.7333   0.0902   0.0744
                  -1.7333    0.0902      1.71
22 VWAAK    -2.6931   -2.5271   -1.2871   -4.7548   3.6521   0.8524
   0.0744   -1.7333   0.0902   0.0744   -1.7333   0.0902   2.8369
                  1.4092    -3.1398      0.04
23 VAAWK    -2.6931   -2.5271   -1.2871   0.0744   -1.7333   0.0902
   0.0744   -1.7333   0.0902   -4.7548   3.6521   0.8524   2.8369
                  1.4092    -3.1398      0.23
24 EKWAP     3.0777   0.3891   -0.0701   2.8369   1.4092   -3.1398
   -4.7548   3.6521   0.8524   0.0744   -1.7333   0.0902   -1.2201
                  0.8829    2.2253      1.3
25 VKWAP    -2.6931   -2.5271   -1.2871   2.8369   1.4092   -3.1398
   -4.7548   3.6521   0.8524   0.0744   -1.7333   0.0902   -1.2201
                  0.8829    2.2253      2.35
26 RKWAP     2.8827   2.5215   -3.4435   2.8369   1.4092   -3.1398
   -4.7548   3.6521   0.8524   0.0744   -1.7333   0.0902   -1.2201
                  0.8829    2.2253      1.98
27 VEWVK    -2.6931   -2.5271   -1.2871   3.0777   0.3891   -0.0701
   -4.7548   3.6521   0.8524   -2.6931   -2.5271   -1.2871   2.8369
                  1.4092    -3.1398      1.71
28 PGFSP    -1.2201   0.8829   2.2253   2.2261   -5.3648   0.3049
   -4.9217   1.2977   0.4473   1.9607   -1.6324   0.5746   -1.2201
                  0.8829    2.2253      0.9
29 FSPFR    -4.9217   1.2977   0.4473   1.9607   -1.6324   0.5746
   -1.2201   0.8829   2.2253   -4.9217   1.2977   0.4473   2.8827
                  2.5215    -3.4435      0.64
30 RYLPT     2.8827   2.5215   -3.4435   -1.3944   2.323   0.0139
   -4.1921  -1.0285   -0.9801   -1.2201   0.8829   2.2253   0.9243
                 -2.0921    -1.3996      0.4
31 GGGGG     2.2261   -5.3648   0.3049   2.2261   -5.3648   0.3049
   2.2261   -5.3648   0.3049   2.2261   -5.3648   0.3049   2.2261
                  -5.3648    0.3049       .
    ;
```

```
data mswkal;
   input n OBSNAM $ v1-v27 LS HA DT @;
   datalines;
1  EM1   2766  2610  3306  3630  3600  3438  3213  3051  2907
   2844  2796  2787  2760  2754  2670  2520  2310  2100  1917  1755
   1602   1467   1353   1260   1167   1101   1017   3.011    0    0
2  EM2    1492   1419   1369   1158   958   887   905   929   920
   887   800   710   617   535   451   368   296   241   190   157
   128   106    89    70    65    56    50    0   0.4005    0
3  EM3   2450  2379  2400  2055  1689  1355  1109  908  750
   673   644   640   630   618   571   512   440   368   305   247
   196   156   120    98    80    61    50    0    0   90.63
4  EM4   2751  2883  3492  3570  3282  2937  2634  2370  2187
   2070  2007  1974  1950  1890  1824  1680  1527  1350  1206  1080
   984   888   810   732   669   630   582   1.482   0.158   40
5  EM5   2652  2691  3225  3285  3033  2784  2520  2340  2235
   2148  2094  2049  2007  1917  1800  1650  1464  1299  1140  1020
   909   810   726   657   594   549   507   1.116   0.4104   30.45
6  EM6   3993  4722  6147  6720  6531  5970  5382  4842  4470
   4200  4077  4008  3948  3864  3663  3390  3090  2787  2481  2241
   2028  1830  1680  1533  1440  1314  1227  3.397  0.3032  50.82
7  EM7   4032  4350  5430  5763  5490  4974  4452  3990  3690
   3474  3357  3300  3213  3147  3000  2772  2490  2220  1980  1779
   1599  1440  1320  1200  1119  1032  957  2.428  0.2981  70.59
8  EM8   4530  5190  6910  7580  7510  6930  6150  5490  4990
   4670  4490  4370  4300  4210  4000  3770  3420  3060  2760  2490
   2230  2060  1860  1700  1590  1490  1380  4.024  0.1153  89.39
9  EM9   4077  4410  5460  5857  5607  5097  4605  4170  3864
   3708  3588  3537  3480  3330  3192  2910  2610  2325  2064  1830
   1638  1476  1350  1236  1122  1044  963  2.275  0.504  81.75
10 EM10  3450  3432  3969  4020  3678  3237  2814  2487  2205
   2061  2001  1965  1947  1890  1776  1635  1452  1278  1128  981
   867   753   663   600   552   507   468   0.9588   0.145   101.1
11 EM11  4989  5301  6807  7425  7155  6525  5784  5166  4695
   4380  4197  4131  4077  3972  3777  3531  3168  2835  2517  2244
   2004  1809  1620  1470  1359  1266  1167  3.19  0.253  120
12 EM12  5340  5790  7590  8390  8310  7670  6890  6190  5700
   5380  5200  5110  5040  4900  4700  4390  3970  3540  3170  2810
   2490  2240  2060  1870  1700  1590  1470  4.132  0.5691  117.7
13 EM13  3162  3477  4365  4650  4470  4107  3717  3432  3228
   3093  3009  2964  2916  2838  2694  2490  2253  2013  1788  1599
   1431  1305  1194  1077  990  927  855  2.16  0.436  27.59
14 EM14  4380  4695  6018  6510  6342  5760  5151  4596  4200
   3948  3807  3720  3672  3567  3438  3171  2880  2571  2280  2046
   1857  1680  1548  1413  1314  1200  1119  3.094  0.2471  61.71
15 EM15  4587  4200  5040  5289  4965  4449  3939  3507  3174
   2970  2850  2814  2748  2670  2529  2328  2088  1851  1641  1431
   1284  1134  1020  918  840  756  714  1.604  0.2856  108.8
16 EM16  4017  4725  6090  6570  6354  5895  5346  4911  4611
   4422  4314  4287  4224  4110  3915  3600  3240  2913  2598  2325
   2088  1917  1734  1587  1452  1356  1257  3.162  0.7012  60
17 EM17  3933  4518  5637  6006  5721  5187  4641  4149  3789
```

```
   3579  3447  3381  3327  3234  3078  2832  2571  2274  2040  1818
   1629   1470   1350   1245   1134   1050   987   2.5   0.3   80
18 EM25  2904  2997  3255  3150  2922  2778  2700  2646  2571
   2487  2370  2250  2127  2052  1713  1419  1200  984  795  648
   525    426    351    291    240    204    162    30    40    150
;
```

```
data fluor5;
   input n OBSNAM $ f1-f30 TOT_LOG TYR_LOG TRY_LOG @;
   datalines;
1   43trp6    -5.91450   -5.91821   -6.90776   -5.42843   -4.11659
   -5.10275   -4.65963   -4.35053   -4.02295   -3.84905   -3.63439
   -3.63439   -3.57199   -3.51325   -3.63439   -3.57199   -3.77226
   -3.77226   -3.84436   -3.93223   -4.01738   -4.02295   -4.11659
   -4.22673   -4.49184   -4.65963   -4.85491   -5.42843   -5.10275
   -5.42843   -6         -8         -6
2   59mix6    -5.90345   -5.90345   -5.90345   -5.08159   -4.21313
   -5.08321   -4.83836   -4.63873   -4.47414   -4.21313   -4.00085
   -4.09835   -4.00085   -4.00085   -3.90704   -4.00085   -4.09835
   -4.09835   -4.20639   -4.09835   -4.21313   -4.21313   -4.33514
   -4.47414   -4.63873   -4.83836   -4.83710   -5.08482   -5.41037
   -5.41037   -6         -7         -6.04576
3   51mix6    -5.90712   -5.90712   -5.41485   -4.84343   -4.21313
   -4.84343   -4.84343   -4.48295   -4.34281   -4.00633   -4.00633
   -3.91202   -3.83044   -3.83044   -3.75502   -3.91202   -4.00633
   -4.00085   -4.21313   -4.21313   -4.33514   -4.48295   -4.48295
   -4.64183   -4.84089   -5.08806   -5.08806   -5.41485   -5.41485
   -5.41485   -6         -6.60206   -6.12494
4   49mix6    -5.41936   -5.09130   -5.09130   -4.64808   -4.00633
   -4.84597   -4.64808   -4.48295   -4.34281   -4.21991   -4.21991
   -4.21991   -4.11047   -4.11047   -4.11047   -4.21991   -4.21991
   -4.34281   -4.48295   -4.48295   -4.65017   -4.65017   -4.84597
   -4.84597   -5.09293   -5.09130   -5.41936   -5.41710   -5.41710
   -5.90712   -6         -6.30103   -6.30103
5   53mix6    -5.08321   -4.83710   -4.83710   -4.47414   -3.82585
   -4.47414   -4.63873   -4.83836   -4.83710   -4.63873   -4.63873
   -4.64080   -4.64080   -4.63873   -4.63873   -4.83710   -4.83836
   -4.83836   -5.08321   -5.08159   -5.08321   -5.41037   -5.41037
   -5.40813   -5.40813   -5.89980   -5.41037   -5.90345   -5.89980
   -6.90776   -6         -6.12494    -6.60206
6   57mix6    -5.08159   -4.83584   -4.63873   -4.47414   -3.82585
   -4.63563   -4.63769   -4.63769   -4.83710   -5.08159   -5.08159
   -5.40813   -5.08159   -5.07999   -5.40813   -5.40813   -5.40813
   -5.40813   -5.40813   -5.40813   -5.40813   -5.89980   -5.89980
   -5.89980   -5.89980   -5.89980   -5.89980   -5.89980   -6.90776
   -6.90776   -6         -6.04576   -7
7   41tyro6   -5.10440   -4.66174   -4.66174   -4.35831   -3.70501
   -4.50081   -4.66174   -4.85877   -5.10440   -5.43071   -5.43299
   -5.91821   -5.91821   -5.91821   -5.43071   -5.91821   -5.91821
   -5.91821   -5.91821   -5.91821   -5.91821   -5.91821   -5.91821
   -6.90776   -5.91821   -5.91821   -6.90776   -6.90776   -5.91821
   -5.91821   -6         -6         -8
8   28trp5    -5.93698   -5.93698   -5.93698   -4.52636   -3.54391
   -3.17009   -2.57308   -2.11470   -1.79234   -1.56446   -1.40039
   -1.30383   -1.24355   -1.21322   -1.23953   -1.29182   -1.37255
   -1.45333   -1.57070   -1.69679   -1.80060   -1.87269   -2.00828
   -2.19804   -2.46864   -2.70605   -2.98975   -3.20893   -3.38434
   -3.60087   -5         -8         -5
9   37mix5    -5.10935   -4.86524   -4.50081   -4.02856   -3.31870
```

```
   -3.07046   -2.56916   -2.20673   -1.89492   -1.68438   -1.51645
   -1.42325   -1.36735   -1.34777   -1.37397   -1.41461   -1.50292
   -1.59613   -1.71786   -1.83898   -1.92730   -1.99746   -2.11752
   -2.33304   -2.56655   -2.87351   -3.10555   -3.31319   -3.57913
   -3.78099   -5         -6         -5.04576
10  33mix5    -4.36615   -4.12894   -3.78099   -3.46734   -3.03655
   -2.93935   -2.59294   -2.26799   -1.98843   -1.79078   -1.64880
   -1.56518   -1.51978   -1.50882   -1.52413   -1.57968   -1.66474
   -1.75776   -1.88217   -2.03707   -2.08998   -2.16204   -2.28435
   -2.46510   -2.76145   -3.03655   -3.27017   -3.51998   -3.70908
   -3.93734   -5         -5.60206   -5.12494
11  31mix5    -3.78981   -3.37261   -3.11903   -2.91507   -2.67075
   -2.71810   -2.55490   -2.39800   -2.22868   -2.08482   -1.97091
   -1.90160   -1.85957   -1.83678   -1.88118   -1.94890   -2.00909
   -2.12746   -2.23036   -2.38055   -2.45457   -2.51331   -2.62417
   -2.82683   -3.11677   -3.37261   -3.58632   -3.78539   -4.03986
   -4.36615   -5         -5.30103   -5.30103
12  35mix5    -3.32146   -2.97006   -2.76462   -2.59428   -2.44646
   -2.54849   -2.61593   -2.61730   -2.57177   -2.54977   -2.50838
   -2.48651   -2.48771   -2.48651   -2.52949   -2.59294   -2.68825
   -2.79198   -2.90772   -3.03655   -3.14888   -3.18932   -3.27280
   -3.46734   -3.78099   -4.02856   -4.24053   -4.50081   -4.66918
   -4.86524   -5         -5.12494   -5.60206
13  39mix5    -3.14191   -2.81175   -2.56395   -2.40351   -2.28102
   -2.50226   -2.58893   -2.70605   -2.84215   -2.96423   -3.06830
   -3.10332   -3.18206   -3.26754   -3.36102   -3.41125   -3.51661
   -3.57555   -3.70501   -3.84905   -3.93223   -3.93223   -4.02856
   -4.23361   -4.50081   -4.66386   -4.86006   -5.10440   -5.43071
   -5.43299   -5         -5.04576   -6
14  26tyro5   -3.03655   -2.69563   -2.46393   -2.32075   -2.23942
   -2.44415   -2.60234   -2.82346   -3.14423   -3.39621   -3.74228
   -4.06285   -4.39816   -4.69948   -4.89285   -5.13790   -5.13961
   -5.46084   -5.46319   -5.94458   -5.46084   -5.13790   -5.13961
   -5.13790   -5.13790   -5.46319   -5.46084   -5.46084   -5.46084
   -5.46084   -5         -5         -8
15  17mix35   -6.21461   -5.80914   -5.11433   -3.96332   -2.89679
   -2.26905   -1.67467   -1.23488   -0.89988   -0.65865   -0.49687
   -0.39499   -0.33528   -0.31482   -0.33327   -0.37679   -0.45271
   -0.54938   -0.65764   -0.79673   -0.87789   -0.95412   -1.06022
   -1.26642   -1.52028   -1.80393   -2.04400   -2.26905   -2.49617
   -2.71357   -4.52288   -8         -4.52288
16  19mix35   -5.51647   -5.29433   -4.82333   -3.85848   -2.82683
   -2.24885   -1.68341   -1.21817   -0.90731   -0.65764   -0.50083
   -0.40046   -0.34455   -0.32288   -0.34172   -0.38656   -0.46080
   -0.55436   -0.66538   -0.80267   -0.88717   -0.95996   -1.07186
   -1.27154   -1.54145   -1.81419   -2.05760   -2.28868   -2.49617
   -2.71206   -4.52288   -6.52288   -4.52724
17  21mix35   -5.51896   -5.29433   -4.50081   -3.86323   -2.82683
   -2.27985   -1.71641   -1.26185   -0.93918   -0.69417   -0.53556
   -0.44429   -0.38413   -0.36852   -0.37726   -0.42065   -0.49458
   -0.59566   -0.70627   -0.82383   -0.91732   -0.98824   -1.10265
   -1.29430   -1.56523   -1.84118   -2.08353   -2.31973   -2.52074
   -2.72876   -4.52288   -6.12494   -4.53387
18  23mix35   -5.29433   -4.70499   -4.26158   -3.60454   -2.72571
```

```
    -2.23933   -1.68142   -1.25026   -0.92541   -0.69720   -0.53357
    -0.43660   -0.38066   -0.35928   -0.36941   -0.42632   -0.49918
    -0.59102   -0.70124   -0.84262   -0.92541   -0.98926   -1.10875
    -1.30978   -1.57876   -1.85189   -2.08957   -2.31567   -2.52074
    -2.74264   -4.52288   -5.82391   -4.54516
19   25mix35   -4.60018   -4.06868   -3.76360   -3.26231   -2.59830
    -2.19122   -1.68040   -1.27314   -0.95795   -0.72904   -0.57281
    -0.47042   -0.42223   -0.40699   -0.42223   -0.46780   -0.53772
    -0.63925   -0.75341   -0.88702   -0.96751   -1.03713   -1.14718
    -1.35658   -1.61858   -1.88552   -2.14064   -2.35916   -2.58495
    -2.79198   -4.52288   -5.52288   -4.56864
20   27mix35   -3.81218   -3.37553   -3.02619   -2.72571   -2.24885
    -1.91916   -1.54145   -1.19799   -0.95112   -0.76402   -0.63925
    -0.57024   -0.52843   -0.52503   -0.54986   -0.60647   -0.68914
    -0.78145   -0.90929   -1.03145   -1.12621   -1.19132   -1.30335
    -1.50260   -1.78379   -2.05760   -2.29660   -2.50715   -2.72723
    -2.97006   -4.52288   -5.12494   -4.64782
21   29mix35   -3.05336   -2.64086   -2.38163   -2.19427   -1.97652
    -1.91332   -1.72760   -1.51567   -1.31651   -1.15757   -1.02882
    -0.96260   -0.91882   -0.91529   -0.93306   -0.98118   -1.05532
    -1.15655   -1.27054   -1.40870   -1.50458   -1.54618   -1.67504
    -1.87987   -2.13962   -2.41463   -2.65498   -2.87884   -3.07478
    -3.31870   -4.52288   -4.82391   -4.82391
22   28mix35   -2.62555   -2.24781   -2.00396   -1.83917   -1.74171
    -1.79078   -1.78576   -1.77190   -1.72760   -1.66633   -1.61863
    -1.59057   -1.57489   -1.57973   -1.61858   -1.67068   -1.75429
    -1.85732   -1.98181   -2.11428   -2.21046   -2.25838   -2.37947
    -2.57046   -2.85771   -3.11677   -3.34671   -3.56843   -3.76360
    -4.01184   -4.52288   -4.64782   -5.12494
23   26mix35   -2.36979   -1.99011   -1.75429   -1.62364   -1.56041
    -1.65475   -1.77190   -1.89885   -1.98181   -2.07449   -2.15684
    -2.21147   -2.26702   -2.31668   -2.36872   -2.46041   -2.54466
    -2.66787   -2.80677   -2.95076   -3.03032   -3.07478   -3.21389
    -3.37553   -3.68489   -3.90704   -4.12894   -4.33514   -4.50081
    -4.59919   -4.52288   -4.56864   -5.52288
24   24mix35   -2.32585   -1.95200   -1.70227   -1.58265   -1.50715
    -1.62877   -1.77090   -1.94498   -2.11528   -2.29660   -2.44761
    -2.58495   -2.69563   -2.80842   -2.91323   -3.03032   -3.16297
    -3.26492   -3.37553   -3.53359   -3.64200   -3.72140   -3.85848
    -4.01184   -4.26158   -4.50081   -4.70389   -4.82208   -4.95615
    -5.29234   -4.52288   -4.54516   -5.82391
25   22mix35   -2.27682   -1.91237   -1.67702   -1.55566   -1.48665
    -1.62979   -1.79078   -1.96933   -2.20328   -2.43726   -2.65498
    -2.84387   -3.03240   -3.21389   -3.37846   -3.50323   -3.64582
    -3.81218   -3.95807   -4.12894   -4.19306   -4.26158   -4.41455
    -4.50081   -4.82333   -5.11101   -5.11267   -5.29433   -5.29035
    -5.29433   -4.52288   -4.53387   -6.12494
26   20mix35   -2.26606   -1.91237   -1.68783   -1.54618   -1.50009
    -1.64010   -1.80090   -2.01141   -2.27682   -2.54466   -2.82346
    -3.09445   -3.37553   -3.57199   -3.81218   -4.01184   -4.26158
    -4.41455   -4.50081   -4.70499   -4.82333   -4.82333   -4.95615
    -5.11101   -5.11101   -5.51647   -5.52396   -5.80582   -5.80582
    -5.80582   -4.52288   -4.52724   -6.52288
27   18mix35   -2.25838   -1.89986   -1.66633   -1.52386   -1.47880
```

```
   -1.62055   -1.80290   -2.04299   -2.30760   -2.62555   -2.89498
   -3.21389   -3.56843   -3.90704   -4.19306   -4.42285   -4.82457
   -5.11101   -5.11101   -5.51647   -5.51647   -5.51647   -5.51647
   -5.80582   -5.80582   -5.80582   -5.80582   -5.80582   -6.20962
   -6.21461   -4.52288   -4.52288   -8
28  trp2      -5.92194   -5.43528   -4.36615   -3.14888   -2.12419
   -1.39235   -0.78025   -0.33607   -.0023427   0.23325    0.39096
    0.49048    0.54040    0.56258    0.54070    0.48846    0.41390
    0.31276    0.20302    0.062505  -0.027813  -0.097227  -0.21480
   -0.41066   -0.67788   -0.95290   -1.20845   -1.41795   -1.65052
   -1.85457   -4         -8         -4
29  mix5     -3.93223    -3.41125   -2.96423   -2.46158   -1.83559
    -1.30800  -0.79573   -0.38976   -0.076017   0.14652    0.29386
     0.39427   0.44635    0.46027    0.44308    0.38913    0.31379
     0.21986   0.098849  -0.033453  -0.12807   -0.19739   -0.30848
    -0.50580  -0.78458   -1.04962   -1.31256   -1.52855   -1.74474
    -1.96962  -4         -5         -4.04576
30  mix4      -2.99573   -2.47932   -2.09867   -1.80272   -1.45921
   -1.12646   -0.76096   -0.42355   -0.14402    0.059966   0.19516
    0.28803    0.33740    0.35417    0.32966    0.27361    0.20620
    0.10508    -.0091618 -0.14767   -0.24188   -0.30620   -0.42355
   -0.62590   -0.89213   -1.17247   -1.42520   -1.63342   -1.87712
   -2.07147   -4         -4.60206   -4.12494
31  mix3      -2.12830   -1.66142   -1.34438   -1.15996   -0.99631
   -0.87745   -0.69621   -0.49467   -0.31308   -0.16542   -0.042031
    0.032467   0.069339   0.078719   0.050408   -.0057666 -0.081633
   -0.17910   -0.29480   -0.43561   -0.52271   -0.58431   -0.70641
   -0.89794   -1.17795   -1.44647   -1.69592   -1.92230   -2.12830
   -2.34968   -4         -4.30103   -4.30103
32  mix6      -1.14031   -0.75713   -0.49745   -0.36199   -0.32937
   -0.41241   -0.51309   -0.64670   -0.77158   -0.87745   -0.95763
   -1.04001   -1.10407   -1.16165   -1.23258   -1.31718   -1.42520
   -1.54253   -1.66142   -1.80436   -1.87712   -1.95921   -2.03370
   -2.24857   -2.50226   -2.73183   -2.96423   -3.14191   -3.31319
   -3.57555   -4         -4.04576      -5
33  tyro2     -1.08069   -0.70970   -0.46975   -0.33675   -0.32674
   -0.43286   -0.60152   -0.84051   -1.11933   -1.42325   -1.74973
   -2.12085   -2.44877   -2.81842   -3.11002   -3.46734   -3.78099
   -4.02856   -4.24053   -4.36615   -4.50081   -4.36615   -4.50081
   -4.50081   -4.66811   -4.66811   -4.86524   -4.86524   -5.10935
   -5.11101   -4         -4         -8
;
```

Appendix 2

Macros

```
   /*************************************************************/
   /*                                                           */
   /* NOTE: These macros work with releases 6.12 and up.        */
   /*       For more information, send e-mail to Bruce          */
   /*       Elsheimer at sasbze@unx.sas.com or Randy Tobias     */
   /*       at sasrdt@unx.sas.com.                              */
   /*                                                           */
   /*************************************************************/

   /*-----------------------------------------------------------------

   DISCLAIMER:

   THIS INFORMATION IS PROVIDED BY SAS INSTITUTE INC. AS  A  SERVICE  TO
   ITS USERS.  IT  IS  PROVIDED  "AS  IS".   THERE  ARE  NO  WARRANTIES,
   EXPRESSED  OR  IMPLIED,  AS  TO  MERCHANTABILITY  OR  FITNESS  FOR  A
   PARTICULAR PURPOSE REGARDING THE ACCURACY OF THE  MATERIALS  OR  CODE
   CONTAINED HEREIN.

   -----------------------------------------------------------------*/
%macro res_plot(ds);

   /************************************************************
   /  Plots Y-residuals vs. predicted values for each PLS      /
   /      component.                                          /
   /  Variable:                                               /
   /      DS -       The input data set:  Must at least       /
   /                 contain variables for observation        /
   /                 numbers, predicted values and residuals  /
   /                 and should not contain missing values.   /
   ************************************************************/

data ds; set &ds;

data _NULL_; set &ds;
  call symput('max_n',n);
run;

%do i=1 %to &num_y;

   axis1 label=(angle=270 rotate=90 "Residual")
         major=(number=5) minor=none;

   axis2 label=("Prediction for Response &i") minor=none;

data res_anno;     *** Annotation Data Set for Plot ***;
```

```
   length text $ %length(&max_n);
   retain function 'label' position '5' hsys '3' xsys '2' ysys '2' ;
   set ds;
   text=%str(n);  x=&predname&i;  y=y&resname&i;
run;

proc gplot data=ds;
   plot y&resname&i*&predname&i/anno=res_anno vaxis=axis1
                                haxis=axis2 vref=0 lvref=2 frame;
   symbol1 v=none i=none;
run;

%end;

%mend;

%macro nor_plot(ds);

   /*************************************************************
   /  Plots Y-residuals vs. Normal quantiles for each PLS      /
   /  component.                                               /
   /  Variable:                                                /
   /      DS -       The input data set:  Must at least        /
   /                 contain variables for observation         /
   /                 numbers, predicted values and residuals   /
   /                 and should not contain missing values.    /
   *************************************************************/

data ds; set &ds;
run;

data _NULL_; set &ds;
  call symput('max_n',n);
run;

%do i=1 %to &num_y;
   data ds; set ds;
      if y&resname&i=. then delete;
run;
%end;

data _NULL_; set ds;
   call symput('numobs',_N_);
run;

%do i=1 %to &num_y;

proc sort data=ds; by y&resname&i;

   /*************************************************************
   /  Calculate the expected values under normality for each  /
   /  residual.                                                /
   *************************************************************/
```

```
data resid&i; set ds(keep=n y&resname&i);
   v=(_n_ - 0.375)/(&numobs+0.25);
   z=probit(v);
run;

axis1 label=(angle=270 rotate=90 "Y&i Residual")
      major=(number=5) minor=none;

axis2 label=('Normal Quantile') minor=none;

data nor_anno;             *** Annotation Data Set for Plot ***;
   length text $ %length(&max_n);
   retain function 'label' position '5' hsys '3' xsys '2' ysys '2' ;
   set resid&i;
   text=%str(n); x=z; y=y&resname&i;
run;

proc gplot data=resid&i;
   plot y&resname&i*z/anno=nor_anno vaxis=axis1 haxis=axis2
                      frame;
   symbol1 v=none i=none;
run;

%end;

%mend;

%macro plot_scr(ds,
                max_lv=&lv);

   /*************************************************************
   /  Plots the Y-scores vs. the corresponding X-scores for    /
   /  each PLS component.                                      /
   /  Variables:                                               /
   /      DS -       The data set containing the scores and   /
   /                 observation numbers.                     /
   /      MAX_LV -   Number of the last PLS component to have /
   /                 its scores plotted.                      /
   *************************************************************/

data dsout; set &ds;      *** Uses nonmissing observations ***;
   if n ^= .;
run;

data _NULL_; set &ds;
  call symput('max_n',n);
run;

%do i=1 %to &max_lv;

data pltanno;           *** Annotation Data Set for Plot ***;
   length text $ %length(&max_n);
   retain function 'label' position '5' hsys '3' xsys '2' ysys '2' ;
   set dsout;
```

```
   text=%str(n); x=&xscrname&i; y=&yscrname&i;
run;

axis1 label=(angle=270 rotate=90 "Y score &i")
      major=(number=5) minor=none;

axis2 label=("X-score &i") minor=none;

proc gplot data=dsout;
   plot &yscrname&i*&xscrname&i/anno=pltanno vaxis=axis1
                                haxis=axis2 frame;
   symbol1 v=none i=none;
run;

%end;

%mend;

%macro plotxscr(ds,
                max_lv=&lv);

   /************************************************************
   /  Plots X-scores for a given number of PLS components      /
   /  vs. those of the preceding PLS component.                /
   /  Variables:                                               /
   /      DS -       The data set containing the X-scores and  /
   /                 observation numbers.                      /
   /      MAX_LV -   Number of the last PLS component to have  /
   /                 its scores plotted.                       /
   ************************************************************/

data dsout; set &ds;
   if n ^= .;               *** Uses nonmissing observations ***;
run;

data _NULL_; set &ds;
  call symput('max_n',n);
run;

%do i=1 %to %eval(&max_lv-1);

%let j=%eval(&i+1);

data pltanno;            *** Annotation Data Set for Plot ***;
   length text $ %length(&max_n);
   retain function 'label' position '5' hsys '3' xsys '2' ysys '2' ;
   set dsout;
   text=%str(n); x=&xscrname&i; y=&xscrname&j;
run;

axis1 label=(angle=270 rotate=90 "X score &j")
      major=(number=5) minor=none;

axis2 label=("X-score &i") minor=none;
```

```
proc gplot data=dsout;
   plot &xscrname&j*&xscrname&i/anno=pltanno vaxis=axis1
                                haxis=axis2 frame;
   symbol1 v=none i=none;
run;

%end;

%mend;

%macro get_wts(dsoutmod,
               dsxwts=xwts);

   /*********************************************************
   /  Gets X-weights w from OUTMODEL data set:               /
   /    1.  Gets appropriate section of OUTMODEL data set.  /
   /    2.  Outputs this data set as DSXWTS1 (will be used  /
   /        in VIP calculation.)                            /
   /    3.  Transposes the data set so the w's are the      /
   /        variables.                                      /
   /    4.  Renames the columns to w1 - wA, where A is the  /
   /        number of PLS components LV in the final model. /
   /  Variables:                                            /
   /      DSOUTMOD - Name of the OUTMODEL data set generated /
   /                 by proc PLS.                          /
   /      DSXWTS -   Name of the data set containing the    /
   /                 X-weights as variables that is output /
   /                 by this macro.                        /
   *********************************************************/

data &dsxwts; set &dsoutmod(keep=_TYPE_ _LV_ &xvars);
   if _TYPE_='WB' then output;

proc transpose data=&dsxwts out=&dsxwts; run;

data &dsxwts; set &dsxwts;
   if _NAME_='_LV_' then delete;
   n=_n_-1;
run;

%do i=1 %to &lv;

data &dsxwts; set &dsxwts;
   rename col&i=w&i;
run;

%end;

%mend;
```

```
%macro plot_wt(ds,
               max_lv=&lv);

   /*************************************************************
   /  Plots X-weights for a given number of PLS components     /
   /  vs. those of the preceding PLS component.                /
   /  Variables:                                               /
   /      DS -       Name of the data set containing the       /
   /                 weights as variables w1-wA, where A=LV,   /
   /                 the number of PLS components, and a       /
   /                 character variable _NAME_ containing the  /
   /                 X-variable names.                         /
   /      MAX_LV -   Number of the last PLS component to have  /
   /                 its weights plotted.                      /
   *************************************************************/

   /*************************************************************
   /  Determine the largest label to be put on plot           /
   *************************************************************/

%let name_len=1;

%do i=1 %to &num_x;
   %let temp=%scan(&xvars,&i,%str( ));
      %if %length(&temp)>&name_len %then %do;
         %let name_len=%length(&temp);
   %end;
%end;

   /*************************************************************
   /  Plot X-weights for each PLS component                    /
   *************************************************************/

%do i=1 %to %eval(&max_lv-1);

   %let j=%eval(&i+1);

data wt_anno;              *** Annotation Data Set for Plot ***;
   length text $ &name_len;
   retain function 'label' position '5' hsys '3' xsys '2' ysys '2' ;
   set &ds;
   text=%str(_name_); x=w&i; y=w&j;
run;

axis1 label=(angle=270 rotate=90 "X weight &j")
      major=(number=5) minor=none;

axis2 label=("X-weight &i") minor=none;

proc gplot data=&ds;
   plot w&j*w&i/anno=wt_anno vaxis=axis1 haxis=axis2 frame;
   symbol1 v=none i=none;
```

```
run;

%end;

%mend;

%macro pltwtfrq(ds,
                plotyvar=W,
                plotxvar=f,
                max_lv=&lv,
                label=Weight);

   /*************************************************************
   /  Plots X-Weights or X-Loadings versus the frequency for   /
   /  spectrometric calibration data sets.                     /
   /  Variables:                                               /
   /      DS -       Data set containing the weights/loadings  /
   /                 as variables with each observation        /
   /                 representing the weights for a particular /
   /                 X-variable, which in this case is a       /
   /                 frequency.                                /
   /      PLOTYVAR - The name (excluding the component number) /
   /                 of the weight/loading variables.  For     /
   /                 example, PLOTYVAR=w if the variables to   /
   /                 be plotted are w1, w2, w3,...             /
   /      PLOTXVAR - The variable name of the frequency        /
   /                 variable.                                 /
   /      MAX_LV -   Number of PLS components to be plotted    /
   /      LABEL -    The label for the vertical axis in the    /
   /                 plot.                                     /
   *************************************************************/

axis1 label=(angle=270 rotate=90 "&label")
   major=(number=5) minor=none;
axis2 label=("Frequency") minor=none;

%let plotvars=%str( );

   %do i=1 %to &max_lv;
      %let plotvars=%str(&plotvars &plotyvar&i);
   %end;

proc gplot data=&ds;
   plot (&plotvars)*&plotxvar/overlay legend vaxis=axis1
                              haxis=axis2 vref=0 lvref=2 frame;
   symbol1 v=none i=spline;
run;

%mend;
```

```
%macro getxload(dsoutmod,
                dsxload=xloads);

   /*********************************************************
   /  Gets X-loadings p from OUTMODEL data set:              /
   /    1.  Gets appropriate section of OUTMODEL data set.  /
   /    2.  Transposes it so the p's are column vectors.    /
   /    3.  Renames the columns to p1 - pA, where A is the  /
   /        number of PLS components in the final model.    /
   /  Variables:                                           /
   /      DSOUTMOD - Name of the OUTMODEL data set produced /
   /                 by proc PLS.                          /
   /      DSXLOAD -  Name of the data set to contain the    /
   /                 X-loadings as variables.              /
   *********************************************************/

data &dsxload; set &dsoutmod(keep=_TYPE_ &xvars);
   if _TYPE_='PQ' then output;

proc transpose data=&dsxload out=&dsxload; run;

data &dsxload; set &dsxload;
   n=_N_;
run;

   %do i=1 %to &lv;

data &dsxload; set &dsxload;
   rename col&i=p&i;
run;

   %end;

%mend;
```

```
%macro pltxload(ds,
                max_lv=&lv);

   /*************************************************************
   /  Plots X-loadings for a given number of PLS components    /
   /  vs. those of the preceding PLS component.                /
   /  Variables:                                               /
   /      DS -       Name of the data set containing the       /
   /                 loadings as variables p1-pA, where A=LV,  /
   /                 the number of PLS components, and a       /
   /                 character variable _NAME_ containing the  /
   /                 X-variable names.                         /
   /      MAX_LV -   Number of the last PLS component to have  /
   /                 its loadings plotted.                     /
   *************************************************************/

   /*************************************************
   /  Determine the largest label to be put on plot           /
   *************************************************/

%let name_len=1;

   %do i=1 %to &num_x;
      %let temp=%scan(&xvars,&i,%str( ));
         %if %length(&temp)>&name_len %then %do;
            %let name_len=%length(&temp);
      %end;
   %end;

      /*************************************************************
      /  Plot X-loadings for each PLS component                    /
      *************************************************************/

   %do i=1 %to %eval(&max_lv - 1);

      %let j=%eval(&i+1);

data pltanno;             *** Annotation Data Set for Plot ***;
   length text $ &name_len;
   retain function 'label' position '5' hsys '3' xsys '2' ysys '2' ;
   set &ds;
   text=%str(_name_); x=p&i; y=p&j;
run;

axis1 label=(angle=270 rotate=90 "X loading &j")
      major=(number=5) minor=none;

axis2 label=("X-loading &i") minor=none;

proc gplot data=&ds;
   plot p&j*p&i/anno=pltanno vaxis=axis1 haxis=axis2 frame;
   symbol1 v=none i=none;
```

```
run;

   %end;

%mend;

%macro getyload(dsoutmod,
                dsyload=yloads);

   /*********************************************************
   / Gets Y-loadings q from OUTMODEL data set:               /
   /    1.  Gets appropriate section of OUTMODEL data set.  /
   /    2.  Transposes it so the q's are column vectors.    /
   /    3.  Renames the columns to q1 - qA, where A is the  /
   /        number of latent variables in the final model.  /
   /  Variables:                                            /
   /      DSOUTMOD - Name of the OUTMODEL data set produced /
   /                 by proc PLS.                          /
   /      DSYLOAD -  Name of the data set to contain the    /
   /                 Y-loadings as variables.              /
   *********************************************************/

data &dsyload; set &dsoutmod(keep=_TYPE_ _LV_ &yvars);
   if _TYPE_='PQ' then output;

proc transpose data=&dsyload out=&dsyload; run;

data &dsyload; set &dsyload;
   if _NAME_='_LV_' then delete;
run;

   %do i = 1 %to &lv;

data &dsyload; set &dsyload;
   rename col&i=q&i;
run;

   %end;

%mend;

%macro plt_y_lv(dsoutmod);

   /*********************************************************
   /  Plots Y-loadings for each Y-variable versus the PLS    /
   /  component.                                            /
   /  Variable:                                             /
   /      DSOUTMOD - The OUTMODEL data set from proc PLS.   /
   *********************************************************/

data dsyload; set &dsoutmod(keep=_TYPE_ _LV_ &yvars);
   if _TYPE_='PQ' then output;

axis1 label=(angle=270 rotate=90 'Y loading')
```

```
      major=(number=5) minor=none;
axis2 label=('PLS Component') order=(1 to &lv by 1) minor=none;

proc gplot data=dsyload;
   plot (&yvars)*_LV_/overlay legend vaxis=axis1 haxis=axis2
                      vref=0 lvref=2 frame;
run;

%mend;

%macro pltyload(ds,
                max_lv=&lv);

      /*************************************************************
      /  Plots Y-loadings for a given number of PLS components    /
      /  vs. those of the preceding PLS component.                /
      /  Variables:                                               /
      /      DS -       Name of the data set containing the       /
      /                 loadings as variables q1-qA, where A=LV,  /
      /                 the number of PLS components, and a       /
      /                 character variable _NAME_ containing the  /
      /                 Y-variable names.                         /
      /      MAX_LV -   Number of the last PLS component to have  /
      /                 its loadings plotted.                     /
      *************************************************************/

      /*************************************************************
      /  Determine the largest label to be put on plot           /
      *************************************************************/

   %let name_len=1;

      %do i=1 %to &num_y;
         %let temp=%scan(&yvars,&i,%str( ));
            %if %length(&temp)>&name_len %then %do;
               %let name_len=%length(&temp);
            %end;
         %end;

         /*************************************************************
         /  Plot Y-loadings for each PLS component                   /
         *************************************************************/

      %do i=1 %to %eval(&max_lv+1);

         %let j=%eval(&i+1);

data pltanno;                *** Annotation Data Set for Plot ***;
   length text $ &name_len;
   retain function 'label' position '5' hsys '3' xsys '2' ysys '2' ;
   set &ds;
   text=%str(_NAME_); x=q&i; y=q&j;
run;
```

```
axis1 label=(angle=270 rotate=90 "Y loading &j")
      major=(number=5) minor=none;

axis2 label=("Y-loading &i") minor=none;

proc gplot data=&ds;
   plot q&j*q&i/anno=pltanno vaxis=axis1 haxis=axis2;
   symbol1 v=none i=none;
run;

      %end;

%mend;

%macro get_bpls(dsoutmod,
                dsout=bpls);

   /************************************************************
   /  Gets B(PLS), the matrix of PLS regression coefficients   /
   /  of Y on X.  For each Y, the values represent the         /
   /  importance of each X-variable in the modeling of the     /
   /  corresponding Y-variable.                                /
   /  Variables:                                               /
   /      DSOUTMOD - Name of the OUTMODEL data set produced    /
   /                 by proc PLS.                             /
   /      DSOUT -    Name of the data set to contain the       /
   /                 regression coefficients, with the        /
   /                 variables representing columns in        /
   /                 B(PLS), and one variable naming the      /
   /                 X-variable for each row of B(PLS).       /
   ************************************************************/

data est_wb; set &dsoutmod;  if _TYPE_='WB' then output; run;
data est_pq; set &dsoutmod;  if _TYPE_='PQ' then output; run;

proc iml;
   use est_wb;
      read all var {&xvars} into w_prime;
      read all var {_Y_} into b;
   use est_pq;
      read all var {&xvars} into p_prime;
      read all var {&yvars} into q_prime;
      W=w_prime';
      P=p_prime';
      Q=q_prime';
      B_PLS = W*inv(P'*W)*diag(b)*Q';
      b_col=('B1':"B&num_y");
      x_var={&xvars};
   create &dsout from B_PLS[colname=b_col rowname=x_var];
   append from B_PLS[rowname=x_var];
      quit;

%mend;
```

```
%macro plt_bpls(ds);

   /*************************************************************
   /  Plot the PLS predictor (regression) coefficients in     /
   /  B(PLS) vs. the frequency, for each response variable.   /
   /  Variables:                                              /
   /      DS -       Data set containing the columns of      /
   /                 B(PLS) as variables, as well as a       /
   /                 variable for the frequency.             /
   *************************************************************/

data &ds; set &ds;
   f=_n_;
run;

   %let plotvars=%str( );

      %do i=1 %to &num_y;
         %let plotvars=%str(&plotvars b&i);
      %end;

axis1 label=(angle=270 rotate=90 'Coefficient')
      major=(number=5) minor=none;
axis2 label=('Frequency') minor=none;

proc gplot data=&ds;
   plot (&plotvars)*f / overlay legend vaxis=axis1 haxis=axis2
                        vref=0 lvref=2 frame;
   symbol1 v=none i=spline;
run;

%mend;

%macro get_vip(dsoutmod,
               dsvip=vip_data);

   /*************************************************************
   /  Calculate VIP:  Variable Importance for the Projection.  /
   /  This represents the importance of each X-variable in     /
   /  the PLS modeling of both the X- and Y-variables.         /
   /  Variables:                                              /
   /      DSOUTMOD - Name of the OUTMODEL data set produced    /
   /                 by proc PLS.                            /
   /      DSVIP -    Name of the data set to contain the      /
   /                 variable named 'VIP' and the names of   /
   /                 X-variables.                            /
   *************************************************************/

data dsxwts; set &dsoutmod(keep=_TYPE_ _LV_ &xvars);
   if _TYPE_='WB' then output;

data y_rsq; set &dsoutmod(keep=_LV_ _TYPE_ &yvars _Y_);
   if _TYPE_='V' then output;
   drop _TYPE_;
```

```
run;

data y_rsq; merge y_rsq dsxwts; by _LV_;
   if _LV_=0 then delete;
run;

proc iml;
   use y_rsq;
      read all var {_Y_} into rsq_y;
      read all var {&xvars} into w_prime;
         A=nrow(rsq_y);
         K=ncol(w_prime);
         W=w_prime';
         Wnorm=W#(1/sqrt(W[##,]));
         part_rsq=rsq_y-(0//rsq_y[1:(A-1),]);
         tot_rsq=rsq_y[A,];
         vip_sq=((Wnorm##2)*part_rsq)#(K/tot_rsq);
         VIP=sqrt(vip_sq);
         x_var={&xvars};
      create &dsvip from VIP[colname='VIP' rowname=x_var];
      append from VIP[rowname=x_var];
      quit;

%mend;

%macro plot_vip(ds);

   /************************************************************
   /  Plot the VIP:  Variable Importance for the Projection.   /
   /  Variables:                                               /
   /      DS -       Data set containing the frequencies       /
   /                 the VIP for each frequency.               /
   ************************************************************/

data &ds; set &ds;
   f=_N_;
run;

axis1 label=(angle=270 rotate=90 'VIP')
      major=(number=10) minor=none;
axis2 label=('Frequency') minor=none;

proc gplot data=&ds;
   plot vip*f / overlay vaxis=axis1 haxis=axis2 vref=0.8 lvref=2
                frame;
   symbol1 v=none i=join;
run;

%mend;
```

```
%macro get_dmod(dsoutput,
                dsdmod=dmod,
                qresname=qres,
                id=n);

   /*************************************************************
   /  Calculate the distance from each data point to the model /
   /  in both the X-space (DMODX) and in the Y-space (DMODY).  /
   /  Variables:                                               /
   /      DSOUTPUT - OUTPUT data set from proc PLS.            /
   /      DSDMOD -   Data set to contain the distances to      /
   /                 the model.                                /
   /      QRESNAME - Suffix of variable names for XQRES and    /
   /                 YQRES assigned by the user in the         /
   /                 proc PLS OUTPUT statement.                /
   /      ID -       Observation identification variable       /
   /                 in input data set.                        /
   *************************************************************/

data trn_out; set &dsoutput;
   if y&qresname ^= . then output;
run;

proc means data=trn_out noprint;
   var xqres;
   output out=outmeans n=n mean=xqres_mn;
run;

data _NULL_; set outmeans;
   call symput('num_trn',n);
   call symput('xqres_mn', xqres_mn);
run;

proc iml;
   use &dsoutput;
   read all var {x&qresname} into xqres;
   read all var {y&qresname} into yqres;
   read all var{&id} into id;
   dmodx=sqrt(xqres/&xqres_mn);
   do i=1 to nrow(xqres);
     if yqres[i]=. then
        dmodx[i]=dmodx[i]/sqrt(&num_trn/(&num_trn-&lv-1));
   end;
   dmody=sqrt(yqres*(&num_trn/(&num_trn-&lv-1)));
   dmodboth=id||dmodx||dmody;
   col={&ID DMODX DMODY};
   create &dsdmod from dmodboth[colname=col];
   append from dmodboth;
quit;

%mend;
```