Overview: PLS Procedure

The PLS procedure fits models by using any one of a number of linear predictive methods, including partial least squares (PLS). Ordinary least squares regression, as implemented in SAS/STAT procedures such as PROC GLM and PROC REG, has the single goal of minimizing sample response prediction error, seeking linear functions of the predictors that explain as much variation in each response as possible. The techniques implemented in the PLS procedure have the additional goal of accounting for variation in the predictors, under the assumption that directions in the predictor space that are well sampled should provide better prediction for new observations when the predictors are highly correlated. All of the techniques implemented in the PLS procedure work by extracting successive linear combinations of the predictors, called factors (also called components, latent vectors, or latent variables), which optimally address one or both of these two goals—explaining response variation and explaining predictor variation. In particular, the method of partial least squares balances the two objectives, seeking factors that explain both response and predictor variation.

Note that the name “partial least squares” also applies to a more general statistical method that is not implemented in this procedure. The partial least squares method was originally developed in the 1960s by the econometrician Herman Wold (1966) for modeling “paths” of causal relation between any number of “blocks” of variables. However, the PLS procedure fits only predictive partial least squares models, with one “block” of predictors and one “block” of responses. If you are interested in fitting more general path models, you should consider using the CALIS procedure.

Basic Features

The techniques implemented by the PLS procedure are as follows:

- principal components regression, which extracts factors to explain as much predictor sample variation as possible
- reduced rank regression, which extracts factors to explain as much response variation as possible. This technique, also known as (maximum) redundancy analysis, differs from multivariate linear regression only when there are multiple responses.
- partial least squares regression, which balances the two objectives of explaining response variation and explaining predictor variation. Two different formulations for partial least squares are available: the original predictive method of Wold (1966) and the SIMPLS method of De Jong (1993).

The number of factors to extract depends on the data. Basing the model on more extracted factors improves the model fit to the observed data, but extracting too many factors can cause overfitting—that is, tailoring the model too much to the current data, to the detriment of future predictions. The PLS procedure enables you to choose the number of extracted factors by cross validation—that is, fitting the model to part of the data, minimizing the prediction error for the unfitted part, and iterating with different portions of the data in the roles of fitted and unfitted. Various methods of cross validation are available, including one-at-a-time validation and splitting the data into blocks. The PLS procedure also offers test set validation, where the model is fit to the entire primary input data set and the fit is evaluated over a distinct test data set.
You can use the general linear modeling approach of the GLM procedure to specify a model for your design, allowing for general polynomial effects as well as classification or ANOVA effects. You can save the model fit by the PLS procedure in a data set and apply it to new data by using the SCORE procedure.

The PLS procedure uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, “Statistical Graphics Using ODS.” For specific information about the statistical graphics available with the PLS procedure, see the PLOTS options in the PROC PLS statements and the section “ODS Graphics” on page 7189.

---

**Getting Started: PLS Procedure**

**Spectrometric Calibration**

The example in this section illustrates basic features of the PLS procedure. The data are reported in Umetrics (1995); the original source is Lindberg, Persson, and Wold (1983). Suppose that you are researching pollution in the Baltic Sea, and you would like to use the spectra of samples of seawater to determine the amounts of three compounds present in samples from the Baltic Sea: lignin sulfonate (ls: pulp industry pollution), humic acids (ha: natural forest products), and optical whitener from detergent (dt). Spectrometric calibration is a type of problem in which partial least squares can be very effective. The predictors are the spectra emission intensities at different frequencies in sample spectrum, and the responses are the amounts of various chemicals in the sample.

For the purposes of calibrating the model, samples with known compositions are used. The calibration data consist of 16 samples of known concentrations of ls, ha, and dt, with spectra based on 27 frequencies (or, equivalently, wavelengths). The following statements create a SAS data set named Sample for these data.

```sas
data Sample;
    input obsnam $ v1-v27 ls ha dt @@;
    datalines;
    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 3.0110 0.0000 0.00
    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 0.0000 0.4005 0.00
    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 0.0000 0.0000 90.63
    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 1.4820 0.1580 40.00
    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 1.1160 0.4104 30.45
    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 3.3970 0.3032 50.82
    EM7 4032 4350 5430 5763 5490 4974 4452 3990 3690 3474 3357
```
Fitting a PLS Model

To isolate a few underlying spectral factors that provide a good predictive model, you can fit a PLS model to the 16 samples by using the following SAS statements:

```sas
proc pls data=sample;
  model ls ha dt = v1-v27;
run;
```

By default, the PLS procedure extracts at most 15 factors. The procedure lists the amount of variation accounted for by each of these factors, both individual and cumulative; this listing is shown in Figure 89.1.
Note that all of the variation in both the predictors and the responses is accounted for by only 15 factors; this is because there are only 16 sample observations. More important, almost all of the variation is accounted for with even fewer factors—one or two for the predictors and three to eight for the responses.

### Selecting the Number of Factors by Cross Validation

A PLS model is not complete until you choose the number of factors. You can choose the number of factors by using cross validation, in which the data set is divided into two or more groups. You fit the model to all groups except one, and then you check the capability of the model to predict responses for the group omitted. Repeating this 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.

To select the number of extracted factors by cross validation, you specify the `CV=` option with an argument that says which cross validation method to use. For example, a common method is split-sample validation, in which the different groups are composed of every $n$th observation beginning with the first, every $n$th observation beginning with the second, and so on. You can use the `CV=SPLIT` option to specify split-sample validation with $n = 7$ by default, as in the following SAS statements:

```sas
proc pls data=sample cv=split;
  model ls ha dt = v1-v27;
run;
```

The resulting output is shown in Figure 89.2 and Figure 89.3.
### Figure 89.2  Split-Sample Validated PRESS Statistics for Number of Factors

**The PLS Procedure**

<table>
<thead>
<tr>
<th>Number of Extracted Factors</th>
<th>Root Mean PRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.107747</td>
</tr>
<tr>
<td>1</td>
<td>0.957983</td>
</tr>
<tr>
<td>2</td>
<td>0.931314</td>
</tr>
<tr>
<td>3</td>
<td>0.520222</td>
</tr>
<tr>
<td>4</td>
<td>0.530501</td>
</tr>
<tr>
<td>5</td>
<td>0.586786</td>
</tr>
<tr>
<td>6</td>
<td>0.475047</td>
</tr>
<tr>
<td>7</td>
<td>0.477595</td>
</tr>
<tr>
<td>8</td>
<td>0.483138</td>
</tr>
<tr>
<td>9</td>
<td>0.485739</td>
</tr>
<tr>
<td>10</td>
<td>0.48946</td>
</tr>
<tr>
<td>11</td>
<td>0.521445</td>
</tr>
<tr>
<td>12</td>
<td>0.525653</td>
</tr>
<tr>
<td>13</td>
<td>0.531049</td>
</tr>
<tr>
<td>14</td>
<td>0.531049</td>
</tr>
<tr>
<td>15</td>
<td>0.531049</td>
</tr>
</tbody>
</table>

Minimum root mean PRESS 0.4750
Minimizing number of factors 6

### Figure 89.3  PLS Variation Summary for Split-Sample Validated Model

<table>
<thead>
<tr>
<th>Number of Extracted Factors</th>
<th>Current</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>97.4607</td>
<td>97.4607</td>
</tr>
<tr>
<td>2</td>
<td>99.6436</td>
<td>24.2435</td>
</tr>
<tr>
<td>3</td>
<td>99.8217</td>
<td>24.5339</td>
</tr>
<tr>
<td>4</td>
<td>99.9414</td>
<td>3.7898</td>
</tr>
<tr>
<td>5</td>
<td>99.9829</td>
<td>1.0045</td>
</tr>
<tr>
<td>6</td>
<td>99.9935</td>
<td>2.2808</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Extracted Factors</th>
<th>Current</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>41.9155</td>
<td>41.9155</td>
</tr>
<tr>
<td>2</td>
<td>66.1590</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>90.6929</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>94.4827</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>95.4873</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>97.7681</td>
<td></td>
</tr>
</tbody>
</table>
The absolute minimum PRESS is achieved with six extracted factors. Notice, however, that this is not much smaller than the PRESS for three factors. By using the CVTEST option, you can perform a statistical model comparison suggested by Van der Voet (1994) to test whether this difference is significant, as shown in the following SAS statements:

```sas
proc pls data=sample cv=split cvtest(seed=12345);
   model ls ha dt = v1-v27;
run;
```

The model comparison test is based on a rerandomization of the data. By default, the seed for this randomization is based on the system clock, but it is specified here. The resulting output is shown in Figure 89.4 and Figure 89.5.

**Figure 89.4 Testing Split-Sample Validation for Number of Factors**

**The PLS Procedure**

<table>
<thead>
<tr>
<th>Number of Extracted Factors</th>
<th>Root Mean PRESS</th>
<th>T^2</th>
<th>Prob &gt; T^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.107747</td>
<td>9.272858</td>
<td>0.0010</td>
</tr>
<tr>
<td>1</td>
<td>0.957983</td>
<td>10.62305</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>2</td>
<td>0.931314</td>
<td>8.950878</td>
<td>0.0010</td>
</tr>
<tr>
<td>3</td>
<td>0.520222</td>
<td>5.133259</td>
<td>0.1440</td>
</tr>
<tr>
<td>4</td>
<td>0.530501</td>
<td>5.168427</td>
<td>0.1340</td>
</tr>
<tr>
<td>5</td>
<td>0.586786</td>
<td>6.437266</td>
<td>0.0150</td>
</tr>
<tr>
<td>6</td>
<td>0.475047</td>
<td>0</td>
<td>1.0000</td>
</tr>
<tr>
<td>7</td>
<td>0.477595</td>
<td>2.809763</td>
<td>0.4750</td>
</tr>
<tr>
<td>8</td>
<td>0.483138</td>
<td>7.189526</td>
<td>0.0110</td>
</tr>
<tr>
<td>9</td>
<td>0.485739</td>
<td>7.931726</td>
<td>0.0070</td>
</tr>
<tr>
<td>10</td>
<td>0.48946</td>
<td>6.612597</td>
<td>0.0150</td>
</tr>
<tr>
<td>11</td>
<td>0.521445</td>
<td>6.666235</td>
<td>0.0130</td>
</tr>
<tr>
<td>12</td>
<td>0.525653</td>
<td>7.092861</td>
<td>0.0080</td>
</tr>
<tr>
<td>13</td>
<td>0.531049</td>
<td>7.538298</td>
<td>0.0030</td>
</tr>
<tr>
<td>14</td>
<td>0.531049</td>
<td>7.538298</td>
<td>0.0030</td>
</tr>
<tr>
<td>15</td>
<td>0.531049</td>
<td>7.538298</td>
<td>0.0030</td>
</tr>
</tbody>
</table>

Minimum root mean PRESS 0.4750
Minimizing number of factors 6
Smallest number of factors with p > 0.1 3
The $p$-value of 0.1430 in comparing the cross validated residuals from models with 6 and 3 factors indicates that the difference between the two models is insignificant; therefore, the model with fewer factors is preferred. The variation summary shows that over 99% of the predictor variation and over 90% of the response variation are accounted for by the three factors.

### Predicting New Observations

Now that you have chosen a three-factor PLS model for predicting pollutant concentrations based on sample spectra, suppose that you have two new samples. The following SAS statements create a data set containing the spectra for the new samples:

```sas
data newobs;
  input obsnam $ v1-v27 @@;
datalines;
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
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;
```

You can apply the PLS model to these samples to estimate pollutant concentration. To do so, append the new samples to the original 16, and specify that the predicted values for all 18 be output to a data set, as shown in the following statements:

```sas
data all;
  set sample newobs;
run;

proc pls data=all nfac=3;
  model ls ha dt = v1-v27;
  output out=pred p=p_ls p_ha p_dt;
run;

proc print data=pred;
  where (obsnam in ('EM17','EM25'));
  var obsnam p_ls p_ha p_dt;
run;
```
The new observations are not used in calculating the PLS model, since they have no response values. Their predicted concentrations are shown in Figure 89.6.

**Figure 89.6** Predicted Concentrations for New Observations

<table>
<thead>
<tr>
<th>Obs</th>
<th>obsnam</th>
<th>p ls</th>
<th>p ha</th>
<th>p dt</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>EM17</td>
<td>2.54261</td>
<td>0.31877</td>
<td>81.4174</td>
</tr>
<tr>
<td>18</td>
<td>EM25</td>
<td>-0.24716</td>
<td>1.37892</td>
<td>46.3212</td>
</tr>
</tbody>
</table>

Finally, if ODS Graphics is enabled, PLS also displays by default a plot of the amount of variation accounted for by each factor, as well as a correlations loading plot that summarizes the first two dimensions of the PLS model. The following statements, which are the same as the previous split-sample validation analysis but with ODS Graphics enabled, additionally produce Figure 89.7 and Figure 89.8:

```plaintext
ods graphics on;
proc pls data=sample cv=split cvtest(seed=12345);
  model ls ha dt = v1-v27;
run;
```

**Figure 89.7** Split-Sample Cross Validation Plot
The cross validation plot in Figure 89.7 gives a visual representation of the selection of the optimum number of factors discussed previously. The correlation loading plot is a compact summary of many features of the PLS model. For example, it shows that the first factor is highly positively correlated with all spectral values, indicating that it is approximately an average of them all; the second factor is positively correlated with the lowest frequencies and negatively correlated with the highest, indicating that it is approximately a contrast between the two ends of the spectrum. The observations, represented by their number in the data set on this plot, are generally spaced well apart, indicating that the data give good information about these first two factors. For more details on the interpretation of the correlation loading plot, see the section “ODS Graphics” on page 7189 and Example 89.1.

The default correlation loading plot for just the first two factors depicts most of the model information. In order to see correlation loadings for all three of the selected factors, you use the NFAC= suboption for the PLOT=CORRLOAD option, as in the following:

```
proc pls data=sample nfac=3 plot=corrload(nfac=3);
   model ls ha dt = v1-v27;
run;
```

The resulting plot is shown in Figure 89.9.
Figure 89.9 Correlation Loading Plot Matrix
Syntax: PLS Procedure

The following statements are available in the PLS procedure. Items within the angle brackets are optional.

```plaintext
PROC PLS <options>;
   BY variables;
   CLASS variables </option>;
   EFFECT name=effect-type (variables</options>);
   ID variables;
   MODEL dependent-variables = effects </options>;
   OUTPUT OUT=SAS-data-set <options>;
```

To analyze a data set, you must use the PROC PLS and MODEL statements. You can use the other statements as needed. CLASS and EFFECT statements, if present, must precede the MODEL statement.

PROC PLS Statement

```plaintext
PROC PLS <options>;
```

The PROC PLS statement invokes the PLS procedure. Optionally, you can also indicate the analysis data and method in the PROC PLS statement. Table 89.1 summarizes the options available in the PROC PLS statement.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENSCALE</td>
<td>Displays the centering and scaling information</td>
</tr>
<tr>
<td>CV=</td>
<td>Specifies the cross validation method to be used</td>
</tr>
<tr>
<td>CVTEST</td>
<td>Specifies that van der Voet’s (1994) randomization-based model comparison test be performed</td>
</tr>
<tr>
<td>DATA=</td>
<td>Names the SAS data set</td>
</tr>
<tr>
<td>DETAILS</td>
<td>Displays the details of the fitted model</td>
</tr>
<tr>
<td>METHOD=</td>
<td>Specifies the general factor extraction method to be used</td>
</tr>
<tr>
<td>MISSING=</td>
<td>Specifies how observations with missing values are to be handled in computing the fit</td>
</tr>
<tr>
<td>NFAC=</td>
<td>Specifies the number of factors to extract</td>
</tr>
<tr>
<td>NOCENTER</td>
<td>Suppresses centering of the responses and predictors before fitting</td>
</tr>
<tr>
<td>NOCVSTDIZE</td>
<td>Suppresses re-centering and rescaling of the responses and predictors when cross-validating</td>
</tr>
<tr>
<td>NOPRINT</td>
<td>Suppresses the normal display of results</td>
</tr>
<tr>
<td>NOSCALE</td>
<td>Suppresses scaling of the responses and predictors before fitting</td>
</tr>
<tr>
<td>PLOTS</td>
<td>Controls the plots produced through ODS Graphics</td>
</tr>
<tr>
<td>VARSINGLE</td>
<td>Specifies that continuous model variables be centered and scaled</td>
</tr>
<tr>
<td>VARSS</td>
<td>Displays the amount of variation accounted for in each response and predictor</td>
</tr>
</tbody>
</table>
The following options are available.

**CENSCALE**
lists the centering and scaling information for each response and predictor.

**CV=ONE**

**CV=SPLIT < (n) >**

**CV=BLOCK < (n) >**

**CV=RANDOM < (cv-random-opts) >**

**CV=TESTSET(SAS-data-set)**
specifies the cross validation method to be used. By default, no cross validation is performed. The method **CV=ONE** requests one-at-a-time cross validation, **CV=SPLIT** requests that every nth observation be excluded, **CV=BLOCK** requests that n blocks of consecutive observations be excluded, **CV=RANDOM** requests that observations be excluded at random, and **CV=TESTSET(SAS-data-set)** specifies a test set of observations to be used for validation (formally, this is called “test set validation” rather than “cross validation”). You can, optionally, specify n for **CV=SPLIT** and **CV=BLOCK**; the default is n = 7. You can also specify the following optional **cv-random-options** in parentheses after the **CV=RANDOM** option:

**NITER=n**
specifies the number of random subsets to exclude. The default value is 10.

**NTEST=n**
specifies the number of observations in each random subset chosen for exclusion. The default value is one-tenth of the total number of observations.

**SEED=n**
specifies an integer used to start the pseudo-random number generator for selecting the random test set. If you do not specify a seed, or specify a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer’s clock.

**CVTEST < (cvtest-options) >**
specifies that van der Voet’s (1994) randomization-based model comparison test be performed to test models with different numbers of extracted factors against the model that minimizes the predicted residual sum of squares; for more information, see the section “Cross Validation” on page 7185. You can also specify the following **cv-test-options** in parentheses after the **CVTEST** option:

**PVAL=n**
specifies the cutoff probability for declaring an insignificant difference. The default value is 0.10.

**STAT=test-statistic**
specifies the test statistic for the model comparison. You can specify either T2, for Hotelling’s $T^2$ statistic, or PRESS, for the predicted residual sum of squares. The default value is T2.

**NSAMP=n**
specifies the number of randomizations to perform. The default value is 1000.
SEED=n
specifies the seed value for randomization generation (the clock time is used by default).

DATA=SAS-data-set
names the SAS data set to be used by PROC PLS. The default is the most recently created data set.

DETAILS
lists the details of the fitted model for each successive factor. The details listed are different for different extraction methods; for more information, see the section “Displayed Output” on page 7187.

METHOD=PLS< (PLS-options ) > | SIMPLS | PCR | RRR
specifies the general factor extraction method to be used. The value PLS requests partial least squares, SIMPLS requests the SIMPLS method of De Jong (1993), PCR requests principal components regression, and RRR requests reduced rank regression. The default is METHOD=PLS. You can also specify the following optional PLS-options in parentheses after METHOD=PLS:

ALGORITHM=NIPALS | SVD | EIG | RLGW
names the specific algorithm used to compute extracted PLS factors. NIPALS requests the usual iterative NIPALS algorithm, SVD bases the extraction on the singular value decomposition of X'Y, EIG bases the extraction on the eigenvalue decomposition of Y'XX'Y, and RLGW is an iterative approach that is efficient when there are many predictors. ALGORITHM=SVD is the most accurate but least efficient approach; the default is ALGORITHM=NIPALS.

MAXITER=n
specifies the maximum number of iterations for the NIPALS and RLGW algorithms. The default value is 200.

EPSILON=n
specifies the convergence criterion for the NIPALS and RLGW algorithms. The default value is $10^{-12}$.

MISSING=NONE | AVG | EM < (EM-options ) >
specifies how observations with missing values are to be handled in computing the fit. The default is MISSING=NONE, for which observations with any missing variables (dependent or independent) are excluded from the analysis. MISSING=AVG specifies that the fit be computed by filling in missing values with the average of the nonmissing values for the corresponding variable. If you specify MISSING=EM, then the procedure first computes the model with MISSING=AVG and then fills in missing values by their predicted values based on that model and computes the model again. For both methods of imputation, the imputed values contribute to the centering and scaling values, and the difference between the imputed values and their final predictions contributes to the percentage of variation explained. You can also specify the following optional EM-options in parentheses after MISSING=EM:

MAXITER=n
specifies the maximum number of iterations for the imputation/fit loop. The default value is 1. If you specify a large value of MAXITER=, then the loop will iterate until it converges (as controlled by the EPSILON= option).
**EPSILON=**
specifies the convergence criterion for the imputation/fit loop. The default value is $10^{-8}$. This option is effective only if you specify a large value for the MAXITER= option.

**NFAC=**
specifies the number of factors to extract. The default is $\min\{15, p, N\}$, where $p$ is the number of predictors (the number of dependent variables for METHOD=RRR) and $N$ is the number of runs (observations). This is probably more than you need for most applications. Extracting too many factors can lead to an overfit model, one that matches the training data too well, sacrificing predictive ability. Thus, if you use the default NFAC= specification, you should also either use the CV= option to select the appropriate number of factors for the final model or consider the analysis to be preliminary and examine the results to determine the appropriate number of factors for a subsequent analysis.

**NOCENTER**
suppresses centering of the responses and predictors before fitting. This is useful if the analysis variables are already centered and scaled. For more information, see the section “Centering and Scaling” on page 7186.

**NOCVSTDIZE**
suppresses re-centering and rescaling of the responses and predictors before each model is fit in the cross validation. For more information, see the section “Centering and Scaling” on page 7186.

**NOPRINT**
suppresses the normal display of results. This is useful when you want only the output statistics saved in a data set. Note that this option temporarily disables the Output Delivery System (ODS); for more information, see Chapter 20, “Using the Output Delivery System.”

**NOSCALE**
suppresses scaling of the responses and predictors before fitting. This is useful if the analysis variables are already centered and scaled. For more information, see the section “Centering and Scaling” on page 7186.

**PLOTS**
controls the plots produced through ODS Graphics. When you specify only one plot-request, you can omit the parentheses from around the plot request. For example:

```plaintext
plots=none
plots=cvplot
plots=(diagnostics cvplot)
plots(unpack)=diagnostics
plots(unpack)=(diagnostics corrload(trace=off))
```

ODS Graphics must be enabled before plots can be requested. For example:

```plaintext
ods graphics on;
proc pls data=pentaTrain;
   model log_RAI = S1-S5 L1-L5 P1-P5;
run;
ods graphics off;
```
For more information about enabling and disabling ODS Graphics, see the section “Enabling and Disabling ODS Graphics” on page 607 in Chapter 21, “Statistical Graphics Using ODS.”

If ODS Graphics is enabled but you do not specify the PLOTS= option, then PROC PLS produces by default a plot of the R-square analysis and a correlation loading plot summarizing the first two factors.

The **global-plot-options** include the following:

**FLIP**
interchanges the X-axis and Y-axis dimensions for the score, weight, and loading plots.

**ONLY**
suppresses the default plots. Only plots specifically requested are displayed.

**UNPACKPANEL**

**UNPACK**
suppresses panning. By default, multiple plots can appear in some output panels. Specify UNPACKPANEL to get each plot in a separate panel. You can specify PLOTS(UNPACKPANEL) to unpack only the default plots. You can also specify UNPACKPANEL as a suboption for certain specific plots, as discussed in the following.

The **plot-requests** include the following:

**ALL**
produces all appropriate plots. You can specify other options with ALL—for example, to request all plots and unpack only the residuals, specify PLOTS=(ALL RESIDUALS(UNPACK)).

**CORRLOAD < (options ) >**
produces a correlation loading plot (default). You can specify the following **options**:

**TRACE=OFF | ON**
controls how points that correspond to the X-loadings are depicted. You can specify the following values:

- **OFF** specifies that all X-loadings be depicted in the plot by their names plotted at the corresponding point on the graph.
- **ON** specifies that the positions for all the X-loadings be depicted by a “trace” through the corresponding points.

By default, TRACE=ON if there are more than 20 predictors, and TRACE=OFF otherwise.

**NFAC=n**
specifies the number of factors for which to display correlation loading plots. By default, NFAC=2, which corresponds to a single plot for the first two factors. If you specify a value of \( n \) greater than 2, then \( n(n-1)/2 \) plots are displayed together in a matrix of correlation loading plots. The maximum number of factors that can be displayed in such a matrix is 8.

**UNPACK**
requests that the \( n(n-1)/2 \) correlation loading plots be produced separately instead of in a matrix. This options has no effect unless the NFAC=n option is also specified, with a value of \( n \) greater than 2.
CVPLOT
produces a cross validation and R-square analysis. This plot requires the CV= option to be specified, and is displayed by default in this case.

DIAGNOSTICS < (UNPACK) >
produces a summary panel of the fit for each dependent variable. The summary by default consists of a panel for each dependent variable, with plots depicting the distribution of residuals and predicted values. You can use the UNPACK suboption to specify that the subplots be produced separately.

DMOD
produces the DMODX, DMODY, and DMODXY plots.

DMODX
produces a plot of the distance of each observation to the X model.

DMODXY
produces plots of the distance of each observation to the X and Y models.

DMODY
produces a plot of the distance of each observation to the Y model.

FIT
produces both the fit diagnostic panel and the ParmProfiles plot.

NONE
suppresses the display of graphics.

PARMPROFILES
produces profiles of the regression coefficients.

SCORES < (UNPACK | FLIP) >
produces the XScores, YScores, XYScores, and DModXY plots. You can use the UNPACK suboption to specify that the subplots for scores be produced separately, and the FLIP option to interchange their default X-axis and Y-axis dimensions.

RESIDUALS < (UNPACK) >
plots the residuals for each dependent variable against each independent variable. Residual plots are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately.

VIP
produces profiles of variable importance factors.

WEIGHTS < (UNPACK | FLIP) >
produces all X and Y loading and weight plots, as well as the VIP plot. You can use the UNPACK suboption to specify that the subplots for weights and loadings be produced separately, and the FLIP option to interchange their default X-axis and Y-axis dimensions.
Chapter 89: The PLS Procedure

**XLOADINGPLOT (UNPACK | FLIP)**
produces a scatter plot matrix of X-loadings against each other. Loading scatter plot matrices are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately, and the FLIP option to interchange the default X-axis and Y-axis dimensions.

**XLOADINGPROFILES**
produces profiles of the X-loadings.

**XSCORES (UNPACK | FLIP)**
produces a scatter plot matrix of X-scores against each other. Score scatter plot matrices are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately, and the FLIP option to interchange the default X-axis and Y-axis dimensions.

**XWEIGHTPLOT (UNPACK | FLIP)**
produces a scatter plot matrix of X-weights against each other. Weight scatter plot matrices are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately, and the FLIP option to interchange the default X-axis and Y-axis dimensions.

**XWEIGHTPROFILES**
produces profiles of the X-weights.

**XYSCORES (UNPACK)**
produces a scatter plot matrix of X-scores against Y-scores. Score scatter plot matrices are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately.

**YSCORES (UNPACK | FLIP)**
produces a scatter plot matrix of Y-scores against each other. Score scatter plot matrices are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately, and the FLIP option to interchange the default X-axis and Y-axis dimensions.

**YWEIGHTPLOT (UNPACK | FLIP)**
produces a scatter plot matrix of Y-weights against each other. Weight scatter plot matrices are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately, and the FLIP option to interchange the default X-axis and Y-axis dimensions.

**VARSCALE**
specifies that continuous model variables be centered and scaled prior to centering and scaling the model effects in which they are involved. The rescaling specified by the VARSCALE option is sometimes more appropriate if the model involves crossproducts between model variables; however, the VARSCALE option still might not produce the model you expect. For more information, see the section “Centering and Scaling” on page 7186.
lists, in addition to the average response and predictor sum of squares accounted for by each successive factor, the amount of variation accounted for in each response and predictor.

**BY Statement**

```plaintext
BY variables;
```

You can specify a BY statement with PROC PLS to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the PLS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

**CLASS Statement**

```plaintext
CLASS variables / TRUNCATE;
```

The CLASS statement names the classification variables to be used in the model. The PLS procedure uses a singular, GLM parameterization for effects involving CLASS variables, as discussed in the section “GLM Parameterization of Classification Variables and Effects” on page 385 in Chapter 19, “Shared Concepts and Topics.” Typical classification variables are Treatment, Sex, Race, Group, and Replication. If you use the CLASS statement, it must appear before the MODEL statement statement.

Classification variables can be either character or numeric. By default, class levels are determined from the entire set of formatted values of the CLASS variables.

**NOTE:** Prior to SAS 9, class levels were determined by using no more than the first 16 characters of the formatted values. To revert to this previous behavior, you can use the TRUNCATE option in the CLASS statement.

In any case, you can use formats to group values into levels. See the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

You can specify the following **option** in the CLASS statement after a slash (/):
**TRUNCATE**

Specifies that class levels should be determined by using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option to revert to the levels as determined in releases prior to SAS 9.

---

**EFFECT Statement**

```
EFFECT name=effect-type (variables< / options>) ;
```

The EFFECT statement enables you to construct special collections of columns for design matrices. These collections are referred to as **constructed effects** to distinguish them from the usual model effects formed from continuous or classification variables, as discussed in the section “GLM Parameterization of Classification Variables and Effects” on page 385 in Chapter 19, “Shared Concepts and Topics.”

The following **effect-types** are available.

- **COLLECTION** specifies a collection effect that defines one or more variables as a single effect with multiple degrees of freedom. The variables in a collection are considered as a unit for estimation and inference.
- **LAG** specifies a classification effect in which the level that is used for a particular period corresponds to the level in the preceding period.
- **MULTIMEMBER | MM** specifies a multimember classification effect whose levels are determined by one or more variables that appear in a CLASS statement.
- **POLYNOMIAL | POLY** specifies a multivariate polynomial effect in the specified numeric variables.
- **SPLINE** specifies a regression spline effect whose columns are univariate spline expansions of one or more variables. A spline expansion replaces the original variable with an expanded or larger set of new variables.

Table 89.2 summarizes the **options** available in the EFFECT statement.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Collection Effects Options</strong></td>
<td></td>
</tr>
<tr>
<td>DETAILS</td>
<td>Displays the constituents of the collection effect</td>
</tr>
<tr>
<td><strong>Lag Effects Options</strong></td>
<td></td>
</tr>
<tr>
<td>DESIGNROLE=</td>
<td>Names a variable that controls to which lag design an observation is assigned</td>
</tr>
<tr>
<td>DETAILS</td>
<td>Displays the lag design of the lag effect</td>
</tr>
<tr>
<td>NLAG=</td>
<td>Specifies the number of periods in the lag</td>
</tr>
<tr>
<td>PERIOD=</td>
<td>Names the variable that defines the period. This option is required.</td>
</tr>
<tr>
<td>WITHIN=</td>
<td>Names the variable or variables that define the group within which each period is defined. This option is required.</td>
</tr>
</tbody>
</table>
Table 89.2  continued

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Multimember Effects Options</strong></td>
<td></td>
</tr>
<tr>
<td>NOEFFECT</td>
<td>Specifies that observations with all missing levels for the multimember variables should have zero values in the corresponding design matrix columns</td>
</tr>
<tr>
<td>WEIGHT=</td>
<td>Specifies the weight variable for the contributions of each of the classification effects</td>
</tr>
<tr>
<td><strong>Polynomial Effects Options</strong></td>
<td></td>
</tr>
<tr>
<td>DEGREE=</td>
<td>Specifies the degree of the polynomial</td>
</tr>
<tr>
<td>MDEGREE=</td>
<td>Specifies the maximum degree of any variable in a term of the polynomial</td>
</tr>
<tr>
<td>STANDARDIZE=</td>
<td>Specifies centering and scaling suboptions for the variables that define the polynomial</td>
</tr>
<tr>
<td><strong>Spline Effects Options</strong></td>
<td></td>
</tr>
<tr>
<td>BASIS=</td>
<td>Specifies the type of basis (B-spline basis or truncated power function basis) for the spline effect</td>
</tr>
<tr>
<td>DEGREE=</td>
<td>Specifies the degree of the spline effect</td>
</tr>
<tr>
<td>KNOTMETHOD=</td>
<td>Specifies how to construct the knots for the spline effect</td>
</tr>
</tbody>
</table>

For more information about the syntax of these `effect-types` and how columns of constructed effects are computed, see the section “EFFECT Statement” on page 395 in Chapter 19, “Shared Concepts and Topics.”

ID Statement

ID variables ;

The ID statement names variables whose values are used to label observations in plots. If you do not specify an ID statement, then each observations is labeled in plots by its corresponding observation number.

MODEL Statement

MODEL response-variables = predictor-effects / options ;

The MODEL statement names the responses and the predictors, which determine the Y and X matrices of the model, respectively. Usually you simply list the names of the predictor variables as the model effects, but you can also use the effects notation of PROC GLM to specify polynomial effects and interactions; for more information, see the section “Specification of Effects” on page 3670 in Chapter 47, “The GLM Procedure.”

The MODEL statement is required. You can specify only one MODEL statement (in contrast to the REG procedure, for example, which allows several MODEL statements in the same PROC REG run).

You can specify the following `options` in the MODEL statement after a slash (/).
INTERCEPT
By default, the responses and predictors are centered; thus, no intercept is required in the model. You can specify the INTERCEPT option to override the default.

SOLUTION
lists the coefficients of the final predictive model for the responses. The coefficients for predicting the centered and scaled responses based on the centered and scaled predictors are displayed, as well as the coefficients for predicting the raw responses based on the raw predictors.

OUTPUT Statement

OUTPUT OUT=SAS-data-set keyword=names <...keyword=names>;

You use the OUTPUT statement to specify a data set to receive quantities that can be computed for every input observation, such as extracted factors and predicted values. The following keywords are available:

- **PREDICTED**: predicted values for responses
- **YRESIDUAL**: residuals for responses
- **XRESIDUAL**: residuals for predictors
- **XSCORE**: extracted factors (X-scores, latent vectors, latent variables, $T$)
- **YSCORE**: extracted responses (Y-scores, $U$)
- **STDX**: standardized (centered and scaled) responses
- **STDY**: standardized (centered and scaled) predictors
- **H**: approximate leverage
- **PRESS**: approximate predicted residuals
- **TSQUARE**: scaled sum of squares of score values
- **STDXSSE**: sum of squares of residuals for standardized predictors
- **STDYSSE**: sum of squares of residuals for standardized responses

Suppose that there are $N_x$ predictors and $N_y$ responses and that the model has $N_f$ selected factors.

- The keywords XRESIDUAL and STDX define an output variable for each predictor, so $N_x$ names are required after each one.
- The keywords PREDICTED, YRESIDUAL, STDY, and PRESS define an output variable for each response, so $N_y$ names are required after each of these keywords.
- The keywords XSCORE and YSCORE specify an output variable for each selected model factor. For these keywords, you provide only one base name, and the variables corresponding to each successive factor are named by appending the factor number to the base name. For example, if $N_f = 3$, then a specification of XSCORE=T would produce the variables T1, T2, and T3.
- Finally, the keywords H, TSQUARE, STDXSSE, and STDYSSE each specify a single output variable, so only one name is required after each of these keywords.
Details: PLS Procedure

Regression Methods

All of the predictive methods implemented in PROC PLS work essentially by finding linear combinations of the predictors (factors) to use to predict the responses linearly. The methods differ only in how the factors are derived, as explained in the following sections.

Partial Least Squares

Partial least squares (PLS) works by extracting one factor at a time. Let \( X = X_0 \) be the centered and scaled matrix of predictors and let \( Y = Y_0 \) be the centered and scaled matrix of response values. The PLS method starts with a linear combination \( t = X_0w \) of the predictors, where \( t \) is called a score vector and \( w \) is its associated weight vector. The PLS method predicts both \( X_0 \) and \( Y_0 \) by regression on \( t \):

\[
\begin{align*}
\hat{X}_0 &= tp', \quad \text{where} \quad p' = (t't)^{-1}t'X_0 \\
\hat{Y}_0 &= tc', \quad \text{where} \quad c' = (t't)^{-1}t'Y_0
\end{align*}
\]

The vectors \( p \) and \( c \) are called the X- and Y-loadings, respectively.

The specific linear combination \( t = X_0w \) is the one that has maximum covariance \( t'u \) with some response linear combination \( u = Y_0q \). Another characterization is that the X- and Y-weights \( w \) and \( q \) are proportional to the first left and right singular vectors of the covariance matrix \( X_0'Y_0 \) or, equivalently, the first eigenvectors of \( X_0'Y_0Y_0'X_0 \) and \( Y_0'X_0X_0'Y_0 \), respectively.

This accounts for how the first PLS factor is extracted. The second factor is extracted in the same way by replacing \( X_0 \) and \( Y_0 \) with the X- and Y-residuals from the first factor:

\[
\begin{align*}
X_1 &= X_0 - \hat{X}_0 \\
Y_1 &= Y_0 - \hat{Y}_0
\end{align*}
\]

These residuals are also called the deflated X and Y blocks. The process of extracting a score vector and deflating the data matrices is repeated for as many extracted factors as are wanted.

SIMPLS

Note that each extracted PLS factor is defined in terms of different X-variables \( X_i \). This leads to difficulties in comparing different scores, weights, and so forth. The SIMPLS method of De Jong (1993) overcomes these difficulties by computing each score \( t_i = Xr_i \) in terms of the original (centered and scaled) predictors \( X \). The SIMPLS X-weight vectors \( r_i \) are similar to the eigenvectors of \( SS' = X'YX \), but they satisfy a different orthogonality condition. The \( r_1 \) vector is just the first eigenvector \( e_1 \) (so that the first SIMPLS score is the same as the first PLS score), but whereas the second eigenvector maximizes

\[
e_1'se_2 \text{ subject to } e_1'e_2 = 0
\]

the second SIMPLS weight \( r_2 \) maximizes

\[
r_1'Sr_2 \text{ subject to } r_1'Xr_2 = t_1't_2 = 0
\]
The SIMPLS scores are identical to the PLS scores for one response but slightly different for more than one response; see De Jong (1993) for details. The X- and Y-loadings are defined as in PLS, but since the scores are all defined in terms of $X$, it is easy to compute the overall model coefficients $B$:

$$
\hat{Y} = \sum_{i} t_i c'_i \\
= \sum_{i} Xr_i c'_i \\
= XB, \text{ where } B = RC'
$$

**Principal Components Regression**

Like the SIMPLS method, principal components regression (PCR) defines all the scores in terms of the original (centered and scaled) predictors $X$. However, unlike both the PLS and SIMPLS methods, the PCR method chooses the X-weights/X-scores without regard to the response data. The X-scores are chosen to explain as much variation in $X$ as possible; equivalently, the X-weights for the PCR method are the eigenvectors of the predictor covariance matrix $X'X$. Again, the X- and Y-loadings are defined as in PLS; but, as in SIMPLS, it is easy to compute overall model coefficients for the original (centered and scaled) responses $Y$ in terms of the original predictors $X$.

**Reduced Rank Regression**

As discussed in the preceding sections, partial least squares depends on selecting factors $t = Xw$ of the predictors and $u = Yq$ of the responses that have maximum covariance, whereas principal components regression effectively ignores $u$ and selects $t$ to have maximum variance, subject to orthogonality constraints. In contrast, reduced rank regression selects $u$ to account for as much variation in the predicted responses as possible, effectively ignoring the predictors for the purposes of factor extraction. In reduced rank regression, the Y-weights $q_i$ are the eigenvectors of the covariance matrix $\hat{Y}'_{LS} \hat{Y}_{LS}$ of the responses predicted by ordinary least squares regression; the X-scores are the projections of the Y-scores $Yq_i$ onto the X space.

**Relationships between Methods**

When you develop a predictive model, it is important to consider not only the explanatory power of the model for current responses, but also how well sampled the predictive functions are, since this affects how well the model can extrapolate to future observations. All of the techniques implemented in the PLS procedure work by extracting successive factors, or linear combinations of the predictors, that optimally address one or both of these two goals—explaining response variation and explaining predictor variation. In particular, principal components regression selects factors that explain as much predictor variation as possible, reduced rank regression selects factors that explain as much response variation as possible, and partial least squares balances the two objectives, seeking for factors that explain both response and predictor variation.

To see the relationships between these methods, consider how each one extracts a single factor from the following artificial data set consisting of two predictors and one response:

```plaintext
data data;
  input x1 x2 y;
datalines;
3.37651 2.30716 0.75615
```

```plaintext
```

Regression Methods

```
0.74193 -0.88845  1.15285
4.18747  2.17373  1.42392
0.96097  0.57301  0.27433
-1.11161 -0.75225 -0.25410
-1.38029 -1.31343 -0.04728
 1.28153 -0.13751  1.00341
-1.39242 -2.03615  0.45518
 0.63741  0.06183  0.40699
-2.52533 -1.23726 -0.91080
 2.44277  3.61077 -0.82590
```

; proc pls data=data nfac=1 method=rrr;
  model y = x1 x2;
run;

proc pls data=data nfac=1 method=pcr;
  model y = x1 x2;
run;

proc pls data=data nfac=1 method=pls;
  model y = x1 x2;
run;

The amount of model and response variation explained by the first factor for each method is shown in Figure 89.10 through Figure 89.12.

**Figure 89.10** Variation Explained by First Reduced Rank Regression Factor

**The PLS Procedure**

<table>
<thead>
<tr>
<th>Percent Variation Accounted for by Reduced Rank Regression Factors</th>
<th>Model Effects</th>
<th>Dependent Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Extracted Factors</td>
<td>Current</td>
<td>Total</td>
</tr>
<tr>
<td>Current</td>
<td>Total</td>
<td>Current</td>
</tr>
<tr>
<td>1</td>
<td>15.0661</td>
<td>15.0661</td>
</tr>
</tbody>
</table>

**Figure 89.11** Variation Explained by First Principal Components Regression Factor

**The PLS Procedure**

<table>
<thead>
<tr>
<th>Percent Variation Accounted for by Principal Components</th>
<th>Model Effects</th>
<th>Dependent Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Extracted Factors</td>
<td>Current</td>
<td>Total</td>
</tr>
<tr>
<td>Current</td>
<td>Total</td>
<td>Current</td>
</tr>
<tr>
<td>1</td>
<td>92.9996</td>
<td>9.3787</td>
</tr>
</tbody>
</table>
Notice that, while the first reduced rank regression factor explains all of the response variation, it accounts for only about 15% of the predictor variation. In contrast, the first principal components regression factor accounts for most of the predictor variation (93%) but only 9% of the response variation. The first partial least squares factor accounts for only slightly less predictor variation than principal components but about three times as much response variation.

Figure 89.13 illustrates how partial least squares balances the goals of explaining response and predictor variation in this case.
The ellipse shows the general shape of the 11 observations in the predictor space, with the contours of increasing \( y \) overlaid. Also shown are the directions of the first factor for each of the three methods. Notice that, while the predictors vary most in the \( x_1 = x_2 \) direction, the response changes most in the orthogonal \( x_1 = -x_2 \) direction. This explains why the first principal component accounts for little variation in the response and why the first reduced rank regression factor accounts for little variation in the predictors. The direction of the first partial least squares factor represents a compromise between the other two directions.

**Cross Validation**

None of the regression methods implemented in the PLS procedure fit the observed data any better than ordinary least squares (OLS) regression; in fact, all of the methods approach OLS as more factors are extracted. The crucial point is that, when there are many predictors, OLS can *overfit* the observed data; biased regression methods with fewer extracted factors can provide better predictability of *future* observations. However, as the preceding observations imply, the quality of the observed data fit cannot be used to choose the number of factors to extract; the number of extracted factors must be chosen on the basis of how well the model fits observations not involved in the modeling procedure itself.

One method of choosing the number of extracted factors is to fit the model to only part of the available data (the *training set*) and to measure how well models with different numbers of extracted factors fit the other part of the data (the *test set*). This is called *test set validation*. However, it is rare that you have enough data to make both parts large enough for pure test set validation to be useful. Alternatively, you can make several different divisions of the observed data into training set and test set. This is called *cross validation*, and there are several different types. In *one-at-a-time* cross validation, the first observation is held out as a single-element test set, with all other observations as the training set; next, the second observation is held out, then the third, and so on. Another method is to hold out successive blocks of observations as test sets—for example, observations 1 through 7, then observations 8 through 14, and so on; this is known as *blocked* validation. A similar method is *split-sample* cross validation, in which successive groups of widely separated observations are held out as the test set—for example, observations \( \{1, 11, 21, \ldots\} \), then observations \( \{2, 12, 22, \ldots\} \), and so on. Finally, test sets can be selected from the observed data randomly; this is known as *random sample* cross validation.

Which validation you should use depends on your data. Test set validation is preferred when you have enough data to make a division into a sizable training set and test set that represent the predictive population well. You can specify that the number of extracted factors be selected by test set validation by using the \( \text{CV=TESTSET(data set)} \) option, where *data set* is the name of the data set containing the test set. If you do not have enough data for test set validation, you can use one of the cross validation techniques. The most common technique is one-at-a-time validation (which you can specify with the \( \text{CV=ONE} \) option or just the \( \text{CV} \) option), unless the observed data are serially correlated, in which case either blocked or split-sample validation might be more appropriate (\( \text{CV=BLOCK} \) or \( \text{CV=SPLIT} \)); you can specify the number of test sets in blocked or split-sample validation with a number in parentheses after the \( \text{CV=} \) option. Note that \( \text{CV=ONE} \) is the most computationally intensive of the cross validation methods, since it requires a recomputation of the PLS model for every input observation. Also, note that using random subset selection with \( \text{CV=RANDOM} \) might lead two different researchers to produce different PLS models on the same data (unless the same seed is used).

Whichever validation method you use, the number of factors chosen is usually the one that minimizes the predicted residual sum of squares (PRESS); this is the default choice if you specify any of the CV methods with PROC PLS. However, often models with fewer factors have PRESS statistics that are only marginally
larger than the absolute minimum. To address this, Van der Voet (1994) has proposed a statistical test for comparing the predicted residuals from different models; when you apply van der Voet’s test, the number of factors chosen is the fewest with residuals that are insignificantly larger than the residuals of the model with minimum PRESS.

To see how van der Voet’s test works, let \( R_{i,jk} \) be the \( j \)th predicted residual for response \( k \) for the model with \( i \) extracted factors; the PRESS statistic is \( \sum_{jk} R_{i,jk}^2 \). Also, let \( i_{\text{min}} \) be the number of factors for which PRESS is minimized. The critical value for van der Voet’s test is based on the differences between squared predicted residuals

\[
D_{i,jk} = R_{i,jk}^2 - R_{i_{\text{min}},jk}^2
\]

One alternative for the critical value is \( C_i = \sum_{jk} D_{i,jk} \), which is just the difference between the PRESS statistics for \( i \) and \( i_{\text{min}} \) factors; alternatively, van der Voet suggests Hotelling’s \( T^2 \) statistic \( C_i = \mathbf{d}_i'\mathbf{S}_i^{-1}\mathbf{d}_i \), where \( \mathbf{d}_i \) is the sum of the vectors \( \mathbf{d}_{i,j} = \{D_{i,j1}, \ldots, D_{i,jN_y}\}' \) and \( \mathbf{S}_i \) is the sum of squares and crossproducts matrix

\[
\mathbf{S}_i = \sum_j \mathbf{d}_{i,j}\mathbf{d}_{i,j}'
\]

Virtually, the significance level for van der Voet’s test is obtained by comparing \( C_i \) with the distribution of values that result from randomly exchanging \( R_{i,jk}^2 \) and \( R_{i_{\text{min}},jk}^2 \). In practice, a Monte Carlo sample of such values is simulated and the significance level is approximated as the proportion of simulated critical values that are greater than \( C_i \). If you apply van der Voet’s test by specifying the CVTEST option, then, by default, the number of extracted factors chosen is the least number with an approximate significance level that is greater than 0.10.

### Centering and Scaling

By default, the predictors and the responses are centered and scaled to have mean 0 and standard deviation 1. Centering the predictors and the responses ensures that the criterion for choosing successive factors is based on how much variation they explain, in either the predictors or the responses or both. (See the section “Regression Methods” on page 7181 for more details on how different methods explain variation.) Without centering, both the mean variable value and the variation around that mean are involved in selecting factors. Scaling serves to place all predictors and responses on an equal footing relative to their variation in the data. For example, if Time and Temp are two of the predictors, then scaling says that a change of \( \text{std(Time)} \) in Time is roughly equivalent to a change of \( \text{std(Tmp)} \) in Temp.

Usually, both the predictors and responses should be centered and scaled. However, if their values already represent variation around a nominal or target value, then you can use the NOCENTER option in the PROC PLS statement to suppress centering. Likewise, if the predictors or responses are already all on comparable scales, then you can use the NOSCALE option to suppress scaling.

Note that, if the predictors involve crossproduct terms, then, by default, the variables are not standardized before standardizing the crossproduct. That is, if the \( i \)th values of two predictors are denoted \( x_{i1}^1 \) and \( x_{i2}^1 \), then the default standardized \( i \)th value of the crossproduct is

\[
\frac{x_{i1}^1 x_{i2}^1 - \text{mean}_j(x_{i1}^1 x_{i2}^1)}{\text{std}_j(x_{i1}^1 x_{i2}^1)}
\]
If you want the crossproduct to be based instead on standardized variables

\[
\frac{x_1^1 - m^1}{s^1} \times \frac{x_2^2 - m^2}{s^2}
\]

where \( m^k = \text{mean}_j(x^k_j) \) and \( s^k = \text{std}_j(x^k_j) \) for \( k = 1, 2 \), then you should use the VARSCALE option in the PROC PLS statement. Standardizing the variables separately is usually a good idea, but unless the model also contains all crossproducts nested within each term, the resulting model might not be equivalent to a simple linear model in the same terms. To see this, note that a model involving the crossproduct of two standardized variables

\[
\frac{x_1^1 - m^1}{s^1} \times \frac{x_2^2 - m^2}{s^2} = x_1^1 x_2^2 \frac{1}{s^1 s^2} - x_1^1 \frac{m^2}{s^1 s^2} - x_2^2 \frac{m^1}{s^1 s^2} + \frac{m^1 m^2}{s^1 s^2}
\]

involves both the crossproduct term and the linear terms for the unstandardized variables.

When cross validation is performed for the number of effects, there is some disagreement among practitioners as to whether each cross validation training set should be retransformed. By default, PROC PLS does so, but you can suppress this behavior by specifying the NOCVSTDIZE option in the PROC PLS statement.

---

**Missing Values**

By default, PROC PLS handles missing values very simply. Observations with any missing independent variables (including all classification variables) are excluded from the analysis, and no predictions are computed for such observations. Observations with no missing independent variables but any missing dependent variables are also excluded from the analysis, but predictions are computed.

However, the MISSING= option in the PROC PLS statement provides more sophisticated ways of modeling in the presence of missing values. If you specify MISSING=AVG or MISSING=EM, then all observations in the input data set contribute to both the analysis and the OUTPUT OUT= data set. With MISSING=AVG, the fit is computed by filling in missing values with the average of the nonmissing values for the corresponding variable. With MISSING=EM, the procedure first computes the model with MISSING=AVG, then fills in missing values with their predicted values based on that model and computes the model again. Alternatively, you can specify MISSING=EM(MAXITER=n) with a large value of \( n \) in order to perform this imputation/fit loop until convergence.

---

**Displayed Output**

By default, PROC PLS displays just the amount of predictor and response variation accounted for by each factor.

If you perform a cross validation for the number of factors by specifying the CV option in the PROC PLS statement, then the procedure displays a summary of the cross validation for each number of factors, along with information about the optimal number of factors.

If you specify the DETAILS option in the PROC PLS statement, then details of the fitted model are displayed for each successive factor. These details for each number of factors include the following:

- the predictor loadings
• the predictor weights
• the response weights
• the coded regression coefficients (for METHOD=SIMPLS, PCR, or RRR)

If you specify the CENSCALE option in the PROC PLS statement, then centering and scaling information for each response and predictor is displayed.

If you specify the VARSS option in the PROC PLS statement, the procedure displays, in addition to the average response and predictor sum of squares accounted for by each successive factor, the amount of variation accounted for in each response and predictor.

If you specify the SOLUTION option in the MODEL statement, then PROC PLS displays the coefficients of the final predictive model for the responses. The coefficients for predicting the centered and scaled responses based on the centered and scaled predictors are displayed, as well as the coefficients for predicting the raw responses based on the raw predictors.

### ODS Table Names

PROC PLS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 89.3. For more information about ODS, see Chapter 20, “Using the Output Delivery System.”

<table>
<thead>
<tr>
<th>ODS Table Name</th>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVResults</td>
<td>Results of cross validation</td>
<td>PROC</td>
<td>CV</td>
</tr>
<tr>
<td>CenScaleParms</td>
<td>Parameter estimates for centered and scaled data</td>
<td>MODEL</td>
<td>SOLUTION</td>
</tr>
<tr>
<td>CodedCoef</td>
<td>Coded coefficients</td>
<td>PROC</td>
<td>DETAILS</td>
</tr>
<tr>
<td>MissingIterations</td>
<td>Iterations for missing value imputation</td>
<td>PROC</td>
<td>MISSING=EM</td>
</tr>
<tr>
<td>ModelInfo</td>
<td>Model information</td>
<td>PROC</td>
<td>default</td>
</tr>
<tr>
<td>NObs</td>
<td>Number of observations</td>
<td>PROC</td>
<td>default</td>
</tr>
<tr>
<td>ParameterEstimates</td>
<td>Parameter estimates for raw data</td>
<td>MODEL</td>
<td>SOLUTION</td>
</tr>
<tr>
<td>PercentVariation</td>
<td>Variation accounted for by each factor</td>
<td>PROC</td>
<td>default</td>
</tr>
<tr>
<td>ResidualSummary</td>
<td>Residual summary from cross validation</td>
<td>PROC</td>
<td>CV</td>
</tr>
<tr>
<td>XEffectCenScale</td>
<td>Centering and scaling information for predictor effects</td>
<td>PROC</td>
<td>CENSCALE</td>
</tr>
<tr>
<td>XLoadings</td>
<td>Loadings for independents</td>
<td>PROC</td>
<td>DETAILS</td>
</tr>
<tr>
<td>XVariableCenScale</td>
<td>Centering and scaling information for predictor variables</td>
<td>PROC</td>
<td>CENSCALE and VARSCALE</td>
</tr>
<tr>
<td>XWeights</td>
<td>Weights for independents</td>
<td>PROC</td>
<td>DETAILS</td>
</tr>
<tr>
<td>YVariableCenScale</td>
<td>Centering and scaling information for responses</td>
<td>PROC</td>
<td>CENSCALE</td>
</tr>
<tr>
<td>YWeights</td>
<td>Weights for dependents</td>
<td>PROC</td>
<td>DETAILS</td>
</tr>
</tbody>
</table>
ODS Graphics

Statistical procedures use ODS Graphics to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, “Statistical Graphics Using ODS.”

Before you create graphs, ODS Graphics must be enabled (for example, by specifying the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section “Enabling and Disabling ODS Graphics” on page 607 in Chapter 21, “Statistical Graphics Using ODS.”

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section “A Primer on ODS Statistical Graphics” on page 606 in Chapter 21, “Statistical Graphics Using ODS.”

When ODS Graphics is enabled, by default the PLS procedure produces a plot of the variation accounted for by each extracted factor, as well as a correlation loading plot for the first two extracted factors (if the final model has at least two factors). The plot of the variation accounted for can take several forms:

- If the PLS analysis does not include cross validation, then the plot shows the total R square for both model effects and the dependent variables against the number of factors.

- If you specify the CV= option to select the number of factors in the final model by cross validation, then the plot shows the R-square analysis discussed previously as well as the root mean PRESS from the cross validation analysis, with the selected number of factors identified by a vertical line.

The correlation loading plot for the first two factors summarizes many aspects of the two most significant dimensions of the model. It consists of overlaid scatter plots of the scores of the first two factors, the loadings of the model effects, and the loadings of the dependent variables. The loadings are scaled so that the amount of variation in the variables that is explained by the model is proportional to the distance from the origin; circles indicating various levels of explained variation are also overlaid on the correlation loading plot. Also, the correlation between the model approximations for any two variables is proportional to the length of the projection of the point corresponding to one variable on a line through the origin passing through the point corresponding to the other variable; the sign of the correlation corresponds to which side of the origin the projected point falls on.

The R square and the first two correlation loadings are plotted by default when ODS Graphics is enabled, but you can produce many other plots for the PROC PLS analysis.

ODS Graph Names

PROC PLS assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 89.4.

<table>
<thead>
<tr>
<th>ODS Graph Name</th>
<th>Plot Description</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>CorrLoadPlot</td>
<td>Correlation loading plot (default)</td>
<td>PLOT=CORRLOAD(option)</td>
</tr>
<tr>
<td>CVPlot</td>
<td>Cross validation and R-square analysis (default, as appropriate)</td>
<td>CV=</td>
</tr>
</tbody>
</table>
Table 89.4  continued

<table>
<thead>
<tr>
<th>ODS Graph Name</th>
<th>Plot Description</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>DModXPPlot</td>
<td>Distance of each observation to the X model</td>
<td>PLOT=DMODX</td>
</tr>
<tr>
<td>DModXYPlot</td>
<td>Distance of each observation to the X and Y models</td>
<td>PLOT=DMODXY</td>
</tr>
<tr>
<td>DModYPPlot</td>
<td>Distance of each observation to the Y model</td>
<td>PLOT=DMODY</td>
</tr>
<tr>
<td>DiagnosticsPlot</td>
<td>Panel of diagnostic plots for the fit</td>
<td>PLOT=DIAGNOSTICS</td>
</tr>
<tr>
<td>AbsResidualByPredicted</td>
<td>Absolute residual by predicted values</td>
<td>PLOT=DIAGNOSTICS(UNPACK)</td>
</tr>
<tr>
<td>ObservedByPredicted</td>
<td>Observed by predicted</td>
<td>PLOT=DIAGNOSTICS(UNPACK)</td>
</tr>
<tr>
<td>QQPlot</td>
<td>Residual Q-Q plot</td>
<td>PLOT=DIAGNOSTICS(UNPACK)</td>
</tr>
<tr>
<td>ResidualByPredicted</td>
<td>Residual by predicted values</td>
<td>PLOT=DIAGNOSTICS(UNPACK)</td>
</tr>
<tr>
<td>ResidualHistogram</td>
<td>Residual histogram</td>
<td>PLOT=DIAGNOSTICS(UNPACK)</td>
</tr>
<tr>
<td>RFPlot</td>
<td>RF plot</td>
<td>PLOT=DIAGNOSTICS(UNPACK)</td>
</tr>
<tr>
<td>ParmProfiles</td>
<td>Profiles of regression coefficients</td>
<td>PLOT=PARMPROFILES</td>
</tr>
<tr>
<td>R2Plot</td>
<td>R-square analysis (default, as appropriate)</td>
<td></td>
</tr>
<tr>
<td>ResidualPlots</td>
<td>Residuals for each dependent variable</td>
<td>PLOT=RESIDUALS</td>
</tr>
<tr>
<td>VariableImportancePlot</td>
<td>Profile of variable importance factors</td>
<td>PLOT=VIP</td>
</tr>
<tr>
<td>XLoadingPlot</td>
<td>Scatter plot matrix of X-loadings against each other</td>
<td>PLOT=XLOADINGPLOT</td>
</tr>
<tr>
<td>XLoadingProfiles</td>
<td>Profiles of the X-loadings</td>
<td>PLOT=XLOADINGPROFILES</td>
</tr>
<tr>
<td>XScorePlot</td>
<td>Scatter plot matrix of X-scores against each other</td>
<td>PLOT=XSCORES</td>
</tr>
<tr>
<td>XWeightPlot</td>
<td>Scatter plot matrix of X-weights against each other</td>
<td>PLOT=XWEIGHTPLOT</td>
</tr>
<tr>
<td>XWeightProfiles</td>
<td>Profiles of the X-weights</td>
<td>PLOT=XWEIGHTPROFILES</td>
</tr>
<tr>
<td>XYScorePlot</td>
<td>Scatter plot matrix of X-scores against Y-scores</td>
<td>PLOT=XYSCORES</td>
</tr>
<tr>
<td>YScorePlot</td>
<td>Scatter plot matrix of Y-scores against each other</td>
<td>PLOT=YSCORES</td>
</tr>
<tr>
<td>YWeightPlot</td>
<td>Scatter plot matrix of Y-weights against each other</td>
<td>PLOT=YWEIGHTPLOT</td>
</tr>
</tbody>
</table>
Example 89.1: Examining Model Details

This example, from Umetrics (1995), demonstrates different ways to examine a PLS model. The data come from the field of drug discovery. New drugs are developed from chemicals that are biologically active. Testing a compound for biological activity is an expensive procedure, so it is useful to be able to predict biological activity from 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 following statements create a data set named pentaTrain, which contains these data.

```sas
data pentaTrain;
  input obsnam $ S1 L1 P1 S2 L2 P2
       S3 L3 P3 S4 L4 P4
       S5 L5 P5 log_RAI @@;
  n = _n_;
  datalines;
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 0.00
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
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 0.20
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
VKAAK -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 0.11
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
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
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
VAAAK -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 -0.10
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
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
```

Examples: PLS Procedure
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 predictive factors that account for most of the variation in the response. Typically, the model is fit for part of the data (the “training” or “work” set), and the quality of the fit is judged by how well it predicts the other part of the data (the “test” or “prediction” set). For this example, the first 15 observations serve as the training set and the rest constitute the test set (see Ufkes et al. 1978, 1982).

When you fit a PLS model, you hope to find a few PLS factors that explain most of the variation in both predictors and responses. Factors that explain response variation provide good predictive models for new responses, and factors that explain predictor variation are well represented by the observed values of the predictors. The following statements fit a PLS model with two factors and save predicted values, residuals, and other information for each data point in a data set named outpls.

```
proc pls data=pentaTrain;
  model log\_RAI = S1-S5 L1-L5 P1-P5;
run;
```

The PLS procedure displays a table, shown in Output 89.1.1, showing how much predictor and response variation is explained by each PLS factor.
Output 89.1.1  Amount of Training Set Variation Explained

The PLS Procedure

<table>
<thead>
<tr>
<th>Number of Extracted Factors</th>
<th>Model Effects</th>
<th>Dependent Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent Variation Accounted for by Partial Least Squares Factors</td>
<td>Current</td>
<td>Total</td>
</tr>
<tr>
<td>1</td>
<td>16.9014</td>
<td>16.9014</td>
</tr>
<tr>
<td>2</td>
<td>12.7721</td>
<td>29.6735</td>
</tr>
<tr>
<td>3</td>
<td>14.6554</td>
<td>44.3289</td>
</tr>
<tr>
<td>4</td>
<td>11.8421</td>
<td>56.1710</td>
</tr>
<tr>
<td>5</td>
<td>10.5894</td>
<td>66.7605</td>
</tr>
<tr>
<td>6</td>
<td>5.1876</td>
<td>71.9481</td>
</tr>
<tr>
<td>7</td>
<td>6.1873</td>
<td>78.1354</td>
</tr>
<tr>
<td>8</td>
<td>7.2252</td>
<td>85.3606</td>
</tr>
<tr>
<td>9</td>
<td>6.7285</td>
<td>92.0891</td>
</tr>
<tr>
<td>10</td>
<td>7.9076</td>
<td>99.9967</td>
</tr>
<tr>
<td>11</td>
<td>0.0033</td>
<td>100.0000</td>
</tr>
<tr>
<td>12</td>
<td>0.0000</td>
<td>100.0000</td>
</tr>
<tr>
<td>13</td>
<td>0.0000</td>
<td>100.0000</td>
</tr>
<tr>
<td>14</td>
<td>0.0000</td>
<td>100.0000</td>
</tr>
<tr>
<td>15</td>
<td>0.0000</td>
<td>100.0000</td>
</tr>
</tbody>
</table>

From Output 89.1.1, note that 97% of the response variation is already explained by just two factors, but only 29% of the predictor variation is explained.

The graphics in PROC PLS, available when ODS Graphics is enabled, make it easier to see features of the PLS model.

If ODS Graphics is enabled, then in addition to the tables discussed previously, PROC PLS displays a graphical depiction of the R-square analysis as well as a correlation loading plot summarizing the model based on the first two PLS factors. The following statements perform the previous analysis with ODS Graphics enabled, producing Output 89.1.2 and Output 89.1.3.

```plaintext
ods graphics on;

proc pls data=pentaTrain;
    model log_RAI = S1-S5 L1-L5 P1-P5;
run;
```
Output 89.1.2 Plot of Proportion of Variation Accounted For

R-Square Analysis

Number of Factors

R Square

- Model Term R Square
- Dependent Variable R Square
Output 89.1.3  Correlation Loading Plot

The plot in Output 89.1.2 of the proportion of variation explained (or R square) makes it clear that there is a plateau in the response variation after two factors are included in the model. The correlation loading plot in Output 89.1.3 summarizes many features of this two-factor model, including the following:

- The X-scores are plotted as numbers for each observation. You should look for patterns or clearly grouped observations. If you see a curved pattern, for example, you might want to add a quadratic term. Two or more groupings of observations indicate that it might be better to analyze the groups separately, perhaps by including classification effects in the model. This plot appears to show most of the observations close together, with a few being more spread out with larger positive X-scores for factor 2. There are no clear grouping patterns, but observation 13 stands out.

- The loadings show how much variation in each variable is accounted for by the first two factors, jointly by the distance of the corresponding point from the origin and individually by the distance for the projections of this point onto the horizontal and vertical axes. That the dependent variable is well explained by the model is reflected in the fact that the point for log_RAI is near the 100% circle.

- You can also use the projection interpretation to relate variables to each other. For example, projecting other variables’ points onto the line that runs through the log_RAI point and the origin, you can see that the PLS approximation for the predictor L3 is highly positively correlated with log_RAI, S3 is somewhat less correlated but in the negative direction, and several predictors including L1, L5, and S5 have very little correlation with log_RAI.
Other graphics enable you to explore more of the features of the PLS model. For example, you can examine the X-scores versus the Y-scores to explore how partial least squares chooses successive factors. 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. When ODS Graphics is enabled, you can plot the X-scores versus the Y-scores by using the PLOT=XYSCORES option, as shown in the following statements.

```sas
proc pls data=pentaTrain nfac=4 plot=XYScores;
    model log_RAI = S1-S5 L1-L5 P1-P5;
run;
```

The plot of the X-scores versus the Y-scores for the first four factors is shown in Output 89.1.4.

---

**Output 89.1.4** X-Scores versus Y-Scores

![X-scores versus Y-scores](image)

For this example, Output 89.1.4 shows high correlation between X- and Y-scores for the first factor but somewhat lower correlation for the second factor and sharply diminishing correlation after that. This adds strength to the judgment that NFAC=2 is the right number of factors for these data and this model. Note that observation 13 is again extreme in the first two plots. This run might be overly influential for the PLS analysis; thus, you should check to make sure it is reliable.
As explained earlier, you can draw some inferences about the relationship between individual predictors and the dependent variable from the correlation loading plot. However, the regression coefficient profile and the variable importance plot give a more direct indication of which predictors are most useful for predicting the dependent variable. The regression coefficients represent the importance each predictor has in the prediction of just the response. The variable importance plot, on the other hand, represents the contribution of each predictor in fitting the PLS model for both predictors and response. It is based on the \textit{Variable Importance for Projection} (VIP) statistic of Wold (1994), which summarizes the contribution a variable makes to the model. If a predictor has a relatively small coefficient (in absolute value) \textit{and} a small value of VIP, then it is a prime candidate for deletion. Wold in Umetrics (1995) considers a value less than 0.8 to be “small” for the VIP. The following statements fit a two-factor PLS model and display these two additional plots.

\begin{verbatim}
    proc pls data=pentaTrain nfac=2 plot=(ParmProfiles VIP);
    model log_RAI = S1-S5 L1-L5 P1-P5;
    run;

    ods graphics off;
\end{verbatim}

The additional graphics are shown in \textbf{Output 89.1.5} and \textbf{Output 89.1.6}.

\textbf{Output 89.1.5}  Variable Importance Plots

\begin{center}
\includegraphics[width=\textwidth]{variable_importance_plot.png}
\end{center}
Chapter 89: The PLS Procedure

Output 89.1.6 Regression Parameter Profile

In these two plots, the variables \( L1 \), \( L2 \), \( P2 \), \( S5 \), \( L5 \), and \( P5 \) have small absolute coefficients and small VIP. Looking back at the correlation loading plot in Output 89.1.2, you can see that these variables tend to be the ones near zero for both PLS factors. You should consider dropping these variables from the model.

Example 89.2: Examining Outliers

This example is a continuation of Example 89.1.

Standard diagnostics for statistical models focus on the response, allowing you to look for patterns that indicate the model is inadequate or for outliers that do not seem to follow the trend of the rest of the data. However, partial least squares effectively models the predictors as well as the responses, so you should consider the pattern of the fit for both. The DModX and DModY statistics give the distance from each point to the PLS model with respect to the predictors and the responses, respectively, and ODS Graphics enables you to plot these values. No point should be dramatically farther from the model than the rest. If there is a group of points that are all farther from the model than the rest, they might have something in common, in which case they should be analyzed separately.
Example 89.2: Examining Outliers

The following statements fit a reduced model to the data discussed in Example 89.1 and plot a panel of standard diagnostics as well as the distances of the observations to the model.

```sas
ods graphics on;

proc pls data=pentaTrain nfac=2 plot=(diagnostics dmod);
  model log_RAI = S1 P1 S2 S3 L3 P3 S4 L4 ;
run;
```

The plots are shown in Output 89.2.1 and Output 89.2.2.

**Output 89.2.1** Model Fit Diagnostics
There appear to be no profound outliers in either the predictor space or the response space.

Example 89.3: Choosing a PLS Model by Test Set Validation

This example demonstrates issues in spectrometric calibration. The data (Umetrics 1995) consist of 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. For example, Figure 89.3.1 shows the observed spectra for three samples, one with only tryptophan, one with only tyrosine, and one with a mixture of the two, all at a total concentration of $10^{-6}$. 
Of the 33 samples, 18 are used as a training set and 15 as a test set. The data originally appear in McAvoy et al. (1989).

These data were created in a lab, with the concentrations fixed in order to provide a wide range of applicability for the model. You want to use a linear function of the logarithms of the spectra to predict the logarithms of tyrosine and tryptophan concentration, as well as the logarithm of the total concentration. Actually, because of the possibility of zeros in both the responses and the predictors, slightly different transformations are used. The following statements create SAS data sets containing the training and test data, named ftrain and ftest, respectively.

```sas
data ftrain;
  input obsnam $ tot tyr f1-f30 @@;
  try = tot - tyr;
  if (tyr) then tyr_log = log10(tyr); else tyr_log = -8;
  if (try) then try_log = log10(try); else try_log = -8;
  tot_log = log10(tot);
  datalines;
  17mix35 0.00003 0
  -6.215 -5.809 -5.114 -3.963 -2.897 -2.269 -1.675 -1.235
  -0.900 -0.659 -0.497 -0.395 -0.335 -0.315 -0.333 -0.377
  -0.453 -0.549 -0.658 -0.797 -0.878 -0.954 -1.060 -1.266
```

---

### Output 89.3.1 Spectra for Three Samples of Tyrosine and Tryptophan

![Graph showing spectra for three samples of Tyrosine and Tryptophan](image-url)
data ftest;
  input obsnam $ tot tyr f1-f30 @@;
  try = tot - tyr;
  if (tyr) then tyr_log = log10(tyr); else tyr_log = -8;
  if (try) then try_log = log10(try); else try_log = -8;
  tot_log = log10(tot);
  datalines;
  43trp6 1E-6 0
  -5.915 -5.918 -6.908 -5.428 -4.117 -5.103 -4.660 -4.351
  -4.492 -4.660 -4.855 -5.428 -5.103 -5.428
  59mix6 1E-6 1E-7
  -5.903 -5.903 -5.903 -5.082 -4.213 -5.083 -4.838 -4.639
  -4.474 -4.213 -4.001 -4.098 -4.001 -4.001 -3.907 -4.001
  -4.639 -4.837 -4.837 -5.085 -5.410 -5.410
  51mix6 1E-6 2.5E-7
  -4.841 -5.088 -5.088 -5.415 -5.415 -5.415
  49mix6 1E-6 5E-7

... more lines ...

... more lines ...

tyro2 0.0001 0.0001
  -1.081 -0.710 -0.470 -0.337 -0.327 -0.433 -0.602 -0.841
  -4.668 -4.668 -4.865 -4.865 -5.109 -5.111
The following statements fit a PLS model with 10 factors.

```sas
proc pls data=ftrain nfac=10;
   model tot_log tyr_log try_log = f1-f30;
run;
```

The table shown in Output 89.3.2 indicates that only three or four factors are required to explain almost all of the variation in both the predictors and the responses.

### Output 89.3.2 Amount of Training Set Variation Explained

#### The PLS Procedure

<table>
<thead>
<tr>
<th>Number of Extracted Factors</th>
<th>Model Effects Current</th>
<th>Total</th>
<th>Dependent Variables Current</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>81.1654</td>
<td>48.3385</td>
<td>16.8113</td>
<td>97.9768</td>
</tr>
<tr>
<td>2</td>
<td>16.8113</td>
<td>32.5465</td>
<td>99.9357</td>
<td>96.1652</td>
</tr>
<tr>
<td>3</td>
<td>1.7639</td>
<td>99.7407</td>
<td>99.7407</td>
<td>92.3289</td>
</tr>
<tr>
<td>4</td>
<td>0.1951</td>
<td>99.9357</td>
<td>3.8363</td>
<td>96.1652</td>
</tr>
<tr>
<td>5</td>
<td>0.0276</td>
<td>99.9634</td>
<td>1.6880</td>
<td>97.8532</td>
</tr>
<tr>
<td>6</td>
<td>0.0132</td>
<td>99.9765</td>
<td>0.7247</td>
<td>98.5779</td>
</tr>
<tr>
<td>7</td>
<td>0.0052</td>
<td>99.9817</td>
<td>0.2926</td>
<td>98.8705</td>
</tr>
<tr>
<td>8</td>
<td>0.0053</td>
<td>99.9870</td>
<td>0.1252</td>
<td>98.9956</td>
</tr>
<tr>
<td>9</td>
<td>0.0049</td>
<td>99.9918</td>
<td>0.1067</td>
<td>99.1023</td>
</tr>
<tr>
<td>10</td>
<td>0.0034</td>
<td>99.9952</td>
<td>0.1684</td>
<td>99.2707</td>
</tr>
</tbody>
</table>

In order to choose the optimal number of PLS factors, you can explore how well models based on the training data with different numbers of factors fit the test data. To do so, use the `CV=TESTSET` option, with an argument pointing to the test data set `ftest`. The following statements also employ the ODS Graphics features in PROC PLS to display the cross validation results in a plot.

```sas
ods graphics on;

proc pls data=ftrain nfac=10 cv=TESTSET(ftest) cvtest(stat=press seed=12345);
   model tot_log tyr_log try_log = f1-f30;
run;
```

The tabular results of the test set validation are shown in Output 89.3.3, and the graphical results are shown in Output 89.3.4. They indicate that, although five PLS factors give the minimum predicted residual sum of squares, the residuals for four factors are insignificantly different from those for five. Thus, the smaller model is preferred.
Output 89.3.3 Test Set Validation for the Number of PLS Factors

The PLS Procedure

<table>
<thead>
<tr>
<th>Number of Extracted Factors</th>
<th>Root Mean PRESS</th>
<th>Prob &gt; PRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3.056797</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>1</td>
<td>2.630561</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>2</td>
<td>1.00706</td>
<td>0.0070</td>
</tr>
<tr>
<td>3</td>
<td>0.664603</td>
<td>0.0020</td>
</tr>
<tr>
<td>4</td>
<td>0.521578</td>
<td>0.3800</td>
</tr>
<tr>
<td>5</td>
<td>0.500034</td>
<td>1.0000</td>
</tr>
<tr>
<td>6</td>
<td>0.513561</td>
<td>0.5100</td>
</tr>
<tr>
<td>7</td>
<td>0.501431</td>
<td>0.6870</td>
</tr>
<tr>
<td>8</td>
<td>1.055791</td>
<td>0.1530</td>
</tr>
<tr>
<td>9</td>
<td>1.435085</td>
<td>0.1010</td>
</tr>
<tr>
<td>10</td>
<td>1.720389</td>
<td>0.0320</td>
</tr>
</tbody>
</table>

Minimum root mean PRESS 0.5000
Minimizing number of factors 5
Smallest number of factors with p > 0.1 4

Percent Variation Accounted for by Partial Least Squares Factors

<table>
<thead>
<tr>
<th>Number of Extracted Factors</th>
<th>Model Effects Current</th>
<th>Total</th>
<th>Dependent Variables Current</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>81.1654</td>
<td>81.1654</td>
<td>48.3385</td>
<td>48.3385</td>
</tr>
<tr>
<td>2</td>
<td>16.8113</td>
<td>97.9768</td>
<td>32.5465</td>
<td>80.8851</td>
</tr>
<tr>
<td>3</td>
<td>1.7639</td>
<td>99.7407</td>
<td>11.4438</td>
<td>92.3289</td>
</tr>
<tr>
<td>4</td>
<td>0.1951</td>
<td>99.9357</td>
<td>3.8363</td>
<td>96.1652</td>
</tr>
</tbody>
</table>
Example 89.3: Choosing a PLS Model by Test Set Validation

Output 89.3.4 Test Set Validation Plot

The factor loadings show how the PLS factors are constructed from the centered and scaled predictors. For spectral calibration, it is useful to plot the loadings against the frequency. In many cases, the physical meanings that can be attached to factor loadings help to validate the scientific interpretation of the PLS model. You can use ODS Graphics with PROC PLS to plot the loadings for the four PLS factors against frequency, as shown in the following statements.

```sas
proc pls data=ftrain nfac=4 plot=XLoadingProfiles;
   model tot_log tyr_log try_log = f1-f30;
run;
ods graphics off;
```

The resulting plot is shown in Output 89.3.5.
Chapter 89: The PLS Procedure

Output 89.3.5 Predictor Loadings across Frequencies

Notice that all four factors handle frequencies below and above about 7 or 8 differently. For example, the first factor is very nearly a simple contrast between the averages of the two sets of frequencies, and the second factor appears to be approximately a weighted sum of only the frequencies in the first set.

Example 89.4: Partial Least Squares Spline Smoothing

The EFFECT statement makes it easy to construct a wide variety of linear models. In particular, you can use the spline effect to add smoothing terms to a model. A particular benefit of using spline effects in PROC PLS is that, when operating on spline basis functions, the partial least squares algorithm effectively chooses the amount of smoothing automatically, especially if you combine it with cross validation for the selecting the number of factors. This example employs the EFFECT statement to demonstrate partial least squares spline smoothing of agricultural data.
Weibe (1935) presents data from a study of uniformity of wheat yields over a certain rectangular plot of land. The following statements read these wheat yield measurements, indexed by row and column distances, into the SAS data set Wheat:

```sas
data Wheat;
    keep Row Column Yield;
    input Yield @@;
    iRow = int((_N_-1)/12);
    iCol = mod( _N_-1 ,12);
    Column = iCol*15 + 1; /* Column distance, in feet */
    Row = iRow* 1 + 1; /* Row distance, in feet */
    Row = 125 - Row + 1; /* Invert rows */
    datalines;
    715 595 580 580 615 610 540 515 557 665 560 612
    770 710 655 675 700 690 565 585 550 574 511 618
    760 715 690 690 655 725 665 640 665 705 644 705
    665 615 685 555 585 630 550 520 553 616 573 570
    755 730 670 580 545 620 580 525 503 599 612
    745 670 585 560 550 710 590 545 538 587 600 664
    645 690 550 520 450 630 535 505 530 536 611 578
    585 495 455 470 445 555 500 450 420 461 531 559
    ... more lines ...
    570 585 635 765 550 675 765 620 608 705 677 660
    505 500 580 655 470 565 570 555 537 585 589 619
    465 430 510 680 460 600 670 615 620 594 616 784
    ;
```

The following statements use the PLS procedure to smooth these wheat yields using two spline effects, one for rows and another for columns, in addition to their crossproduct. Each spline effect has, by default, seven basis columns; thus their crossproduct has $49 = 7^2$ columns, for a total of 63 parameters in the full linear model. However, the predictive PLS model does not actually need to have 63 degrees of freedom. Rather, the degree of smoothing is controlled by the number of PLS factors, which in this case is chosen automatically by random subset validation with the CV=RANDOM option.

```sas
ods graphics on;
proc pls data=Wheat cv=random(seed=1) cvtest(seed=12345)
    plot(only)=contourfit(obs=gradient);
    effect splCol = spline(Column);
    effect splRow = spline(Row );
    model Yield = splCol|splRow;
run;
```

These statements produce the output shown in Output 89.4.1 through Output 89.4.4.
Output 89.4.1  Default Spline Basis: Model and Data Information

The PLS Procedure

<table>
<thead>
<tr>
<th>Data Set</th>
<th>WORK.WHEAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor Extraction Method</td>
<td>Partial Least Squares</td>
</tr>
<tr>
<td>PLS Algorithm</td>
<td>NIPALS</td>
</tr>
<tr>
<td>Number of Response Variables</td>
<td>1</td>
</tr>
<tr>
<td>Number of Predictor Parameters</td>
<td>63</td>
</tr>
<tr>
<td>Missing Value Handling</td>
<td>Exclude</td>
</tr>
<tr>
<td>Maximum Number of Factors</td>
<td>15</td>
</tr>
<tr>
<td>Validation Method</td>
<td>10-fold Random Subset Validation</td>
</tr>
<tr>
<td>Random Subset Seed</td>
<td>1</td>
</tr>
<tr>
<td>Validation Testing Criterion</td>
<td>Prob T**2 &gt; 0.1</td>
</tr>
<tr>
<td>Number of Random Permutations</td>
<td>1000</td>
</tr>
<tr>
<td>Random Permutation Seed</td>
<td>12345</td>
</tr>
</tbody>
</table>

Number of Observations Read 1500
Number of Observations Used 1500

Output 89.4.2  Default Spline Basis: Random Subset Validated PRESS Statistics for Number of Factors

<table>
<thead>
<tr>
<th>Number of Extracted Factors</th>
<th>Root Mean PRESS</th>
<th>T**2</th>
<th>Prob &gt; T**2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.066355</td>
<td>251.8793</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>1</td>
<td>0.826177</td>
<td>123.8161</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>2</td>
<td>0.745877</td>
<td>61.6035</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>3</td>
<td>0.725181</td>
<td>44.99644</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>4</td>
<td>0.701464</td>
<td>23.20199</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>5</td>
<td>0.687164</td>
<td>8.369711</td>
<td>0.0030</td>
</tr>
<tr>
<td>6</td>
<td>0.683917</td>
<td>8.775847</td>
<td>0.0010</td>
</tr>
<tr>
<td>7</td>
<td>0.677969</td>
<td>2.907019</td>
<td>0.0830</td>
</tr>
<tr>
<td>8</td>
<td>0.676423</td>
<td>2.190871</td>
<td>0.1340</td>
</tr>
<tr>
<td>9</td>
<td>0.676966</td>
<td>3.191284</td>
<td>0.0600</td>
</tr>
<tr>
<td>10</td>
<td>0.675026</td>
<td>1.334638</td>
<td>0.2390</td>
</tr>
<tr>
<td>11</td>
<td>0.673906</td>
<td>0.556455</td>
<td>0.4470</td>
</tr>
<tr>
<td>12</td>
<td>0.673653</td>
<td>1.257292</td>
<td>0.2790</td>
</tr>
<tr>
<td>13</td>
<td>0.672669</td>
<td>0</td>
<td>1.0000</td>
</tr>
<tr>
<td>14</td>
<td>0.673596</td>
<td>2.386014</td>
<td>0.1190</td>
</tr>
<tr>
<td>15</td>
<td>0.672828</td>
<td>0.02962</td>
<td>0.8820</td>
</tr>
</tbody>
</table>

Minimum root mean PRESS 0.6727
Minimizing number of factors 13
Smallest number of factors with p > 0.1 8
Output 89.4.3  Default Spline Basis: PLS Variation Summary for Split-Sample Validated Model

<table>
<thead>
<tr>
<th>Number of Extracted Factors</th>
<th>Model Effects</th>
<th>Dependent Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>Total</td>
<td>Current</td>
</tr>
<tr>
<td>1</td>
<td>11.5269</td>
<td>40.2471</td>
</tr>
<tr>
<td>2</td>
<td>7.2314</td>
<td>18.7583</td>
</tr>
<tr>
<td>3</td>
<td>6.9147</td>
<td>25.6730</td>
</tr>
<tr>
<td>4</td>
<td>3.8433</td>
<td>29.5163</td>
</tr>
<tr>
<td>5</td>
<td>6.4795</td>
<td>35.9958</td>
</tr>
<tr>
<td>6</td>
<td>7.6201</td>
<td>43.6159</td>
</tr>
<tr>
<td>7</td>
<td>7.3214</td>
<td>50.9373</td>
</tr>
<tr>
<td>8</td>
<td>4.8363</td>
<td>55.7736</td>
</tr>
</tbody>
</table>

Output 89.4.4  Default Spline Basis: Smoothed Yield

Contour Fit Plot for Yield
The cross validation results in Output 89.4.2 point to a model with eight PLS factors; this is the smallest model whose predicted residual sum of squares (PRESS) is insignificantly different from the model with the absolute minimum PRESS. The variation summary in Output 89.4.3 shows that this model accounts for about 60% of the variation in the Yield values. The OBS=GRADIENT suboption for the PLOT=CONTOURFIT option specifies that the observations in the resulting plot, Output 89.4.4, be colored according to the same scheme as the surface of predicted yield. This coloration enables you to easily tell which observations are above the surface of predicted yield and which are below.

The surface of predicted yield is somewhat smoother than what Weibe (1935) settled on originally, with a predominance of simple, elliptically shaped contours. You can easily specify a potentially more granular model by increasing the number of knots in the spline bases. Even though the more granular model increases the number of predictor parameters, cross validation can still protect you from overfitting the data. The following statements are the same as those shown before, except that the spline effects now have twice as many basis functions:

```plaintext
ods graphics on;

proc pls data=Wheat cv=random(seed=1) cvtest(seed=12345) plot(only)=contourfit(obs=gradient);
   effect splCol = spline(Column / knotmethod=equal(14));
   effect splRow = spline(Row / knotmethod=equal(14));
   model Yield = splCol|splRow;
run;
```

The resulting output is shown in Output 89.4.5 through Output 89.4.8.

**Output 89.4.5** More Granular Spline Basis: Model and Data Information

**The PLS Procedure**

<table>
<thead>
<tr>
<th>Data Set</th>
<th>WORK.WHEAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor Extraction Method</td>
<td>Partial Least Squares</td>
</tr>
<tr>
<td>PLS Algorithm</td>
<td>NIPALS</td>
</tr>
<tr>
<td>Number of Response Variables</td>
<td>1</td>
</tr>
<tr>
<td>Number of Predictor Parameters</td>
<td>360</td>
</tr>
<tr>
<td>Missing Value Handling</td>
<td>Exclude</td>
</tr>
<tr>
<td>Maximum Number of Factors</td>
<td>15</td>
</tr>
<tr>
<td>Validation Method</td>
<td>10-fold Random Subset Validation</td>
</tr>
<tr>
<td>Random Subset Seed</td>
<td>1</td>
</tr>
<tr>
<td>Validation Testing Criterion</td>
<td>Prob T**2 &gt; 0.1</td>
</tr>
<tr>
<td>Number of Random Permutations</td>
<td>1000</td>
</tr>
<tr>
<td>Random Permutation Seed</td>
<td>12345</td>
</tr>
</tbody>
</table>

**Number of Observations Read** 1500

**Number of Observations Used** 1500
Output 89.4.6  More Granular Spline Basis: Random Subset Validated PRESS Statistics for Number of Factors

<table>
<thead>
<tr>
<th>Number of Extracted Factors</th>
<th>Root Mean PRESS</th>
<th>T**2</th>
<th>Prob &gt; T**2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.066355</td>
<td>247.9268</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>1</td>
<td>0.652658</td>
<td>20.68658</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>2</td>
<td>0.615087</td>
<td>0.074822</td>
<td>0.7740</td>
</tr>
<tr>
<td>3</td>
<td>0.614128</td>
<td>0</td>
<td>1.0000</td>
</tr>
<tr>
<td>4</td>
<td>0.615268</td>
<td>0.197678</td>
<td>0.6490</td>
</tr>
<tr>
<td>5</td>
<td>0.618001</td>
<td>1.372038</td>
<td>0.2340</td>
</tr>
<tr>
<td>6</td>
<td>0.622949</td>
<td>5.035504</td>
<td>0.0180</td>
</tr>
<tr>
<td>7</td>
<td>0.626482</td>
<td>7.296797</td>
<td>0.0080</td>
</tr>
<tr>
<td>8</td>
<td>0.633316</td>
<td>13.66045</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>9</td>
<td>0.635239</td>
<td>16.16922</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>10</td>
<td>0.636938</td>
<td>18.02295</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>11</td>
<td>0.636494</td>
<td>16.9881</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>12</td>
<td>0.63682</td>
<td>16.83341</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>13</td>
<td>0.637719</td>
<td>16.74157</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>14</td>
<td>0.637627</td>
<td>15.79342</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>15</td>
<td>0.638431</td>
<td>16.12327</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

Minimum root mean PRESS 0.6141
Minimizing number of factors 3
Smallest number of factors with p > 0.1 2

Output 89.4.7  More Granular Spline Basis: PLS Variation Summary for Split-Sample Validated Model

<table>
<thead>
<tr>
<th>Number of Extracted Factors</th>
<th>Percent Variation Accounted for by Partial Least Squares Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Model Effects</td>
</tr>
<tr>
<td>1</td>
<td>1.7967</td>
</tr>
<tr>
<td>2</td>
<td>1.3719</td>
</tr>
</tbody>
</table>
Output 89.4.5 shows that the model now has 360 parameters, many more than before. In Output 89.4.6 you can see that with more granular spline effects, fewer PLS factors are required—only two, in fact. However, Output 89.4.7 shows that this model now accounts for over 70% of the variation in the Yield values, and the contours of predicted values in Output 89.4.8 are less inclined to be simple elliptical shapes.

References


Subject Index

components
  PLS procedure, 7160
constructed effects
  PLS procedure, 7178
cross validation
  PLS procedure, 7171, 7185
factors
  PLS procedure, 7160
latent variables
  PLS procedure, 7160
latent vectors
  PLS procedure, 7160
ODS examples
  PLS procedure, 7205
ODS graph names
  PLS procedure, 7189
options summary
  EFFECT statement, 7178
partial least squares, 7160, 7181
PLS procedure
  algorithms, 7172
  centering, 7186
  compared to other procedures, 7160
  components, 7160
  computation method, 7172
  constructed effects, 7178
  cross validation, 7160, 7185
  cross validation method, 7171
  examples, 7191
  factors, 7160
  factors, selecting the number of, 7163
  introductory example, 7161
  latent variables, 7160
  latent vectors, 7160
  missing values, 7172
  ODS graph names, 7189
  ODS table names, 7188
  outlier detection, 7198
  output data sets, 7180
  output keywords, 7180
  partial least squares regression, 7160, 7181
  predicting new observations, 7166
  principal components regression, 7160, 7182
  reduced rank regression, 7160, 7182
  reduced rank regression, 7160
  PLS procedure, 7182
  reduced rank regression, 7160
  PLS procedure, 7182
regression
  partial least squares (PROC PLS), 7160, 7181
  principal components (PROC PLS), 7160, 7182
  reduced rank (PROC PLS), 7160, 7182
SIMPLS method
  PLS procedure, 7181
test set validation
  PLS procedure, 7185
variable importance for projection, 7197
VIP, 7197
spline smoothing, 7206
test set validation, 7185, 7200
principal components regression (PLS), 7160, 7182
partial least squares regression, 7160
PLS procedure, 7182
variable importance for projection, 7197
VIP, 7197
Syntax Index

ALGORITHM= option
PROC PLS statement, 7172

BY statement
   PLS procedure, 7177

CENSCALE option
PROC PLS statement, 7171

CLASS statement
   PLS procedure, 7177

CV= option
PROC PLS statement, 7171

CVTEST= option
PROC PLS statement, 7171

DATA= option
PROC PLS statement, 7172

DETAILS option
PROC PLS statement, 7172

EFFECT statement
   PLS procedure, 7178

EPSILON= option
PROC PLS statement, METHOD=PLS option, 7172
PROC PLS statement, MISSING=EM option, 7173

ID statement
   PLS procedure, 7179

INTERCEPT option
   MODEL statement (PLS), 7180

MAXITER= option
PROC PLS statement, METHOD=PLS option, 7172
PROC PLS statement, MISSING=EM option, 7173

METHOD= option
PROC PLS statement, 7172

MISSING= option
PROC PLS statement, 7172

MODEL statement
   PLS procedure, 7179

NFAC= option
PROC PLS statement, 7173

NITER= option
PROC PLS statement, 7171

NOCENTER option

PROC PLS statement, 7173

PROC PLS statement, 7173

NOCVSTDIZE option
PROC PLS statement, 7173

NOPRINT option
PROC PLS statement, 7173

NOSCALE option
PROC PLS statement, 7173, 7177

NTEST= option
PROC PLS statement, 7171

OUTPUT statement
   PLS procedure, 7180

PLOTS= option
PROC PLS statement, 7173

PLS procedure
   syntax, 7170

PLS procedure, BY statement, 7177

PLS procedure, CLASS statement, 7177
   TRUNCATE option, 7178

PLS procedure, EFFECT statement, 7178

PLS procedure, ID statement, 7179

PLS procedure, MODEL statement, 7179
   INTERCEPT option, 7180
   SOLUTION option, 7180

PLS procedure, OUTPUT statement, 7180

PLS procedure, PROC PLS statement, 7170
   ALGORITHM= option, 7172
   CENSCALE option, 7171
   CV= option, 7171
   CVTEST= option, 7171
   DATA= option, 7172
   DETAILS option, 7172
   METHOD= option, 7172
   MISSING= option, 7172
   METHOD= option, 7172
   MISSING= option, 7172
   NFAC= option, 7173
   NITER= option, 7171
   NOCENTER option, 7173
   NOCVSTDIZE option, 7173
   NOPRINT option, 7173
   NOSCALE option, 7173, 7177
   NTEST= option, 7171
   PLOTS= option, 7173
   PVAL= option, 7171
   SEED= option, 7171, 7172
   STAT= option, 7171
   VARSCALE option, 7176

PLS procedure, PROC PLS statement, METHOD=PLS option
EPSILON= option, 7172
MAXITER= option, 7172

PLS procedure, PROC PLS statement, MISSING=EM
  option
    EPSILON= option, 7173
    MAXITER= option, 7172
PROC PLS statement, see PLS procedure
PVAL= option
  PROC PLS statement, 7171

SEED= option
  PROC PLS statement, 7171, 7172

SOLUTION option
  MODEL statement (PLS), 7180

STAT= option
  PROC PLS statement, 7171

TRUNCATE option
  CLASS statement (PLS), 7178

VARSCALE option
  PROC PLS statement, 7176