SAS/STAT® 14.3 User's Guide

The HPPLS Procedure

This document is an individual chapter from *SAS/STAT® 14.3 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *SAS/STAT® 14.3 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 14.3 User's Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

September 2017

Chapter 59
The HPPLS Procedure

Contents

Overview: HPPLS Procedure

The HPPLS procedure is a high-performance version of the PLS procedure in SAS/STAT software, which 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 the GLM and REG procedures, has the single goal of minimizing sample response prediction error, and it seeks linear functions of the predictors that explain as much variation in each response as possible. The HPPLS procedure implements techniques that 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 the techniques that the HPPLS procedure implements 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 by seeking factors that explain both response and predictor variation.

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 HPPLS procedure fits only *predictive* partial least squares models that have 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.

PROC HPPLS runs in either single-machine mode or distributed mode.

NOTE: Distributed mode requires SAS High-Performance Statistics.

PROC HPPLS Features

The main features of the HPPLS procedure are as follows:

- supports GLM and reference parameterization for classification effects

- permits any degree of interaction effects that involve classification and continuous variables
- supports partitioning of data into training and testing roles
- supports test set validation to choose the number of extracted factors, where the model is fit to only part of the available data (the training set) and the fit is evaluated over the other part of the data (the test set)
- produces an output data set that contains predicted values and other observationwise statistics

The HPPLS procedure implements the following techniques:

- 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 straightforward implementation of a statistically inspired modification of the partial least squares (SIMPLS) method of De Jong (1993).

Because the HPPLS procedure is a high-performance analytical procedure, it also does the following:

- enables you to run in distributed mode on a cluster of machines that distribute the data and the computations when you license SAS High-Performance Statistics
- enables you to run in single-machine mode on the server where SAS is installed
- exploits all the available cores and concurrent threads, regardless of execution mode

For more information, see the section "Processing Modes" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

PROC HPPLS Contrasted with PROC PLS

The HPPLS procedure and the PLS procedure have the following similarities and differences:

- All the general factor extraction methods that are available in PROC PLS are supported by PROC HPPLS.
- The RLGW algorithm, which is available in PROC PLS to compute extracted PLS factors, is not supported by PROC HPPLS.
- PROC PLS can specify various methods to be used for cross validation. PROC HPPLS supports test set validation only by using the PARTITION statement.

- The CLASS statement in PROC HPPLS permits two parameterizations: the GLM-type parameterization and a reference parameterization. The HPPLS procedure does not mix parameterizations across the variables in the CLASS statement. In other words, all classification variables are in the same parameterization, and this parameterization is either the GLM or reference parameterization. In PROC PLS, only the GLM-type parameterization is supported.

- The HPPLS procedure does not support the EFFECT statement, the MISSING= option, the VARSCALE option, and the PLOTS option that are available in PROC PLS.

- The syntax of the OUTPUT statement in the HPPLS procedure is different from the syntax of the OUTPUT statement in PROC PLS. In the HPPLS procedure, you do not need to provide a prefix in the OUTPUT statement. A default prefix is used if you do not provide one. If you do not specify any output statistics in the OUTPUT statement in PROC HPPLS, the output data set includes the predicted values for response variables. Furthermore, although the OUTPUT statement in the PLS procedure includes the input and BY variables in the output data by default, PROC HPPLS does not include them by default so that it can avoid data duplication for large data sets. In order to include any input or BY variables in the output data set, you must list these variables in the ID statement.

- The HPPLS procedure is primarily designed to operate in the high-performance distributed environment for large-data tasks. By default, PROC HPPLS performs computations on multiple threads. The PLS procedure executes on a single thread.

Getting Started: HPPLS Procedure

Spectrometric Calibration

The example in this section illustrates basic features of the HPPLS procedure. The data are reported in Umetrics (1995); the original source is Lindberg, Persson, and Wold (1983). Suppose you are researching pollution in the Baltic Sea and you want to use the spectra of samples of seawater to determine the amounts of three compounds present in seawater samples: 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 a sample spectrum, and the responses are the amounts of various chemicals in the sample.

For the purposes of calibrating the model, samples that have 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. In order to demonstrate the use of test set validation, the data set contains a variable Role that is used to assign observations to the training and testing roles. In this case, the training role has nine samples and the testing role has seven samples.

```
data Sample;
   input obsnam $ v1-v27 ls ha dt Role $5. @@;
   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 TRAIN
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 TEST
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 TRAIN
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 TEST
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 TEST
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 TRAIN
EM7   4032 4350 5430 5763 5490 4974 4452 3990 3690 3474 3357
      3300 3213 3147 3000 2772 2490 2220 1980 1779 1599 1440
      1320 1200 1119 1032  957         2.4280  0.2981  70.59 TRAIN
EM8   4530 5190 6910 7580 7510 6930 6150 5490 4990 4670 4490
      4370 4300 4210 4000 3770 3420 3060 2760 2490 2230 2060
      1860 1700 1590 1490 1380         4.0240  0.1153  89.39 TRAIN
EM9   4077 4410 5460 5857 5607 5097 4605 4170 3864 3708 3588
      3537 3480 3330 3192 2910 2610 2325 2064 1830 1638 1476
      1350 1236 1122 1044  963         2.2750  0.5040  81.75 TEST
EM10  3450 3432 3969 4020 3678 3237 2814 2487 2205 2061 2001
      1965 1947 1890 1776 1635 1452 1278 1128  981  867  753
       663  600  552  507  468         0.9588  0.1450 101.10 TRAIN
EM11  4989 5301 6807 7425 7155 6525 5784 5166 4695 4380 4197
      4131 4077 3972 3777 3531 3168 2835 2517 2244 2004 1809
      1620 1470 1359 1266 1167         3.1900  0.2530 120.00 TRAIN
EM12  5340 5790 7590 8390 8310 7670 6890 6190 5700 5380 5200
      5110 5040 4900 4700 4390 3970 3540 3170 2810 2490 2240
      2060 1870 1700 1590 1470         4.1320  0.5691 117.70 TEST
EM13  3162 3477 4365 4650 4470 4107 3717 3432 3228 3093 3009
      2964 2916 2838 2694 2490 2253 2013 1788 1599 1431 1305
      1194 1077  990  927  855         2.1600  0.4360  27.59 TRAIN
EM14  4380 4695 6018 6510 6342 5760 5151 4596 4200 3948 3807
      3720 3672 3567 3438 3171 2880 2571 2280 2046 1857 1680
      1548 1413 1314 1200 1119         3.0940  0.2471  61.71 TRAIN
EM15  4587 4200 5040 5289 4965 4449 3939 3507 3174 2970 2850
      2814 2748 2670 2529 2328 2088 1851 1641 1431 1284 1134
      1020  918  840  756  714         1.6040  0.2856 108.80 TEST
EM16  4017 4725 6090 6570 6354 5895 5346 4911 4611 4422 4314
      4287 4224 4110 3915 3600 3240 2913 2598 2325 2088 1917
      1734 1587 1452 1356 1257         3.1620  0.7012  60.00 TEST
;
```

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:

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

By default, the HPPLS procedure extracts at most 15 factors. The default output from this analysis is presented in Figure 59.1 through Figure 59.3.

Figure 59.1 displays the “Performance Information,” “Data Access Information,” and “Model Information” tables.

The “Performance Information” table shows that PROC HPPLS executes in single-machine mode—that is, the model is fit on the machine where the SAS session executes. This run of the HPPLS procedure was performed on a multicore machine that has four CPUs; one computational thread was spawned per CPU.

The “Data Access Information” table shows that the input data set is accessed with the V9 (base) engine on the client machine where the MVA SAS session executes.

The “Model Information” table identifies the data source and shows that the factor extraction method is partial least squares regression (which is the default) and that the nonlinear iterative partial least squares (NIPALS) algorithm (which is also the default) is used to compute extracted PLS factors.

Figure 59.1 Performance, Data Access, and Model Information

The HPPLS Procedure

Performance Information

Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

Data	Engine	Role	Path
WORK.SAMPLE	V9	Input	On Client

Model Information

Data Source	WORK.SAMPLE
Factor Extraction Method	Partial Least Squares
PLS Algorithm	NIPALS
Validation Method	None

Figure 59.2 displays the “Number of Observations” and “Dimensions” tables. The “Number of Observations” table shows that all 16 of the sample observations in the input data are used in the analysis because all samples contain complete data. The “Dimensions” table shows the number of dependent variables, the number of effects, the number of predictor parameters, and the number of factors to extract.

Figure 59.2 Number of Observations and Dimensions

Number of Observations Read	16
Number of Observations Used	16

Figure 59.2 *continued*

Dimensions	
Number of Response Variables	3
Number of Effects	27
Number of Predictor Parameters	27
Number of Factors	15

Figure 59.3 lists the amount of variation, both individual and cumulative, that is accounted for by each of the 15 factors. All the variation in both the predictors and the responses is accounted for by only 15 factors because there are only 16 sample observations. More important, almost all the variation is accounted for with even fewer factors—one or two for the predictors and three to eight for the responses.

Figure 59.3 PLS Variation Summary

Percent Variation Accounted for by Partial Least Squares Factors

	Model Effects		Dependent Variables	
Number of Extracted Factors	**Current**	**Total**	**Current**	**Total**
1	97.46068	97.46068	41.91546	41.91546
2	2.18296	99.64365	24.24355	66.15900
3	0.17806	99.82170	24.53393	90.69293
4	0.11973	99.94143	3.78978	94.48271
5	0.04146	99.98289	1.00454	95.48725
6	0.01058	99.99347	2.28084	97.76809
7	0.00168	99.99515	1.16935	98.93744
8	0.00097586	99.99613	0.50410	99.44153
9	0.00142	99.99755	0.12292	99.56446
10	0.00097037	99.99852	0.11027	99.67472
11	0.00032725	99.99884	0.15227	99.82699
12	0.00029338	99.99914	0.12907	99.95606
13	0.00024792	99.99939	0.03121	99.98727
14	0.00042742	99.99981	0.00651	99.99378
15	0.00018639	100.00000	0.00622	100.00000

Selecting the Number of Factors by Test Set Validation

A PLS model is not complete until you choose the number of factors. You can choose the number of factors by using test set validation, in which the data set is divided into two groups called the training data and test data. You fit the model to the training data, and then you check the capability of the model to predict responses for the test data. The predicted residual sum of squares (PRESS) statistic is based on the residuals that are generated by this process.

To select the number of extracted factors by test set validation, you use the PARTITION statement to specify how observations in the input data set are logically divided into two subsets for model training and testing. For example, you can designate a variable in the input data set and a set of formatted values of that variable to determine the role of each observation, as in the following SAS statements:

```
proc hppls data=sample;
   model ls ha dt = v1-v27;
   partition roleVar = Role(train='TRAIN' test='TEST');
run;
```

The resulting output is shown in Figure 59.4 through Figure 59.6.

Figure 59.4 Model Information and Number of Observations with Test Set Validation

The HPPLS Procedure

Model Information	
Data Source	WORK.SAMPLE
Factor Extraction Method	Partial Least Squares
PLS Algorithm	NIPALS
Validation Method	Test Set Validation

Number of Observations Read	16
Number of Observations Used	16
Number of Observations Used for Training	9
Number of Observations Used for Testing	7

Figure 59.5 Test-Set-Validated PRESS Statistics for Number of Factors

The HPPLS Procedure

Test Set Validation for the Number of Extracted Factors	
Number of Extracted Factors	**Root Mean PRESS**
0	1.426362
1	1.276694
2	1.181752
3	0.656999
4	0.43457
5	0.420916
6	0.585031
7	0.576586
8	0.563935
9	0.563935

Minimum Root Mean PRESS	0.420916
Minimizing Number of Factors	5

Figure 59.6 PLS Variation Summary for Test-Set-Validated Model

Percent Variation Accounted for by Partial Least Squares Factors

	Model Effects		Dependent Variables	
Number of Extracted Factors	**Current**	**Total**	**Current**	**Total**
1	95.92495	95.92495	37.27071	37.27071
2	3.86407	99.78903	32.38167	69.65238
3	0.10170	99.89073	20.76882	90.42120
4	0.08979	99.98052	4.66666	95.08787
5	0.01142	99.99194	3.88184	98.96971

In Figure 59.4, the “Model Information” table indicates that test set validation is used. The “Number of Observations” table shows that nine sample observations are assigned for training roles and seven are assigned for testing roles.

Figure 59.5 provides details about the results from test set validation. These results show that the absolute minimum PRESS is achieved with five 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 that is suggested by Van der Voet (1994) to test whether this difference is significant, as shown in the following SAS statements:

```
proc hppls data=sample cvtest(pval=0.15 seed=12345);
   model ls ha dt = v1-v27;
   partition roleVar = Role(train='TRAIN' test='TEST');
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 presented in Figure 59.7 through Figure 59.9.

Figure 59.7 Model Information with Model Comparison Test

The HPPLS Procedure

Model Information	
Data Source	WORK.SAMPLE
Factor Extraction Method	Partial Least Squares
PLS Algorithm	NIPALS
Validation Method	Test Set Validation
Validation Testing Criterion	Prob T**2 > 0.15
Number of Random Permutations	1000
Random Number Seed for Permutation	12345

Figure 59.8 Testing Test Set Validation for Number of Factors

The HPPLS Procedure

Test Set Validation for the Number of Extracted Factors

Number of Extracted Factors	Root Mean PRESS	T**2	Prob > T**2
0	1.426362	5.191629	0.0650
1	1.276694	6.174825	<.0001
2	1.181752	4.60203	0.0780
3	0.656999	3.09999	0.5200
4	0.43457	4.980227	0.0970
5	0.420916	0	1.0000
6	0.585031	2.05496	0.7420
7	0.576586	3.009172	0.4960
8	0.563935	2.416635	0.7500
9	0.563935	2.416635	0.7490

Minimum Root Mean PRESS	0.420916
Minimizing Number of Factors	5
Smallest Number of Factors with p > 0.15	3

Figure 59.9 PLS Variation Summary for Tested Test-Set-Validated Model

Percent Variation Accounted for by Partial Least Squares Factors

	Model Effects		Dependent Variables	
Number of Extracted Factors	**Current**	**Total**	**Current**	**Total**
1	95.92495	95.92495	37.27071	37.27071
2	3.86407	99.78903	32.38167	69.65238
3	0.10170	99.89073	20.76882	90.42120

The "Model Information" table in Figure 59.7 displays information about the options that are used in the model comparison test. In Figure 59.8, the *p*-value in comparing the test-set validated residuals from models that have five and three factors indicates that the difference between the two models is insignificant; therefore, the model with fewer factors is preferred. The variation summary in Figure 59.9 shows that more than 99% of the predictor variation and more than 90% of the response variation are accounted for by the three factors.

Predicting New Observations

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

```
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 by appending the new samples to the original 16 and specifying that the predicted values for all 18 be output to a data set, as shown in the following statements:

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

proc hppls data=all nfac=2;
   model ls ha dt = v1-v27;
   partition roleVar = Role(train='TRAIN' test='TEST');
   output out=result pred=p;
   id obsnam;
run;

proc print data=result;
   where (obsnam in ('EM17','EM25'));
   var obsnam p_ls p_ha p_dt;
run;
```

The ID statement lists the variable obsnam from the input data set that is transferred to the output data set. The new observations are not used in calculating the PLS model because they have no response values. Their predicted concentrations are shown in Figure 59.10.

Figure 59.10 Predicted Concentrations for New Observations

Obs	obsnam	p_ls	p_ha	p_dt
17	EM17	2.63326	0.22343	80.2027
18	EM25	0.69865	0.14308	98.9937

Syntax: HPPLS Procedure

The following statements are available in the HPPLS procedure:

PROC HPPLS < *options* > ;
  BY *variables* ;
  CLASS *variable* < **(***options***)** >. . . < *variable* < **(***options***)** > > < / *global-options* > ;
  MODEL *response-variables* **=** *predictor-effects* < / *options* > ;
  OUTPUT < **OUT=***SAS-data-set*>
    < *keyword* < **=***prefix* > >. . . < *keyword* < **=***prefix* > > ;
  PARTITION < *partition-options* > ;
  PERFORMANCE < *performance-options* > ;
  ID *variables* ;

The PROC HPPLS statement and a single MODEL statement are required. All other statements are optional. The CLASS statement can appear multiple times. If a CLASS statement is specified, it must precede the MODEL statement. The following sections describe the PROC HPPLS statement and then describe the other statements in alphabetical order.

PROC HPPLS Statement

PROC HPPLS < *options* > ;

The PROC HPPLS statement invokes the HPPLS procedure. Table 59.1 summarizes the options available in the PROC HPPLS statement.

Table 59.1 PROC HPPLS Statement Options

Option	Description
Basic Options	
DATA=	Specifies the input data set
NAMELEN=	Limits the length of effect names
Model Fitting Options	
CVTEST	Requests that van der Voet’s (1994) randomization-based model comparison test be performed
METHOD=	Specifies the general factor extraction method to be used
NFAC=	Specifies the number of factors to extract
NOCENTER	Suppresses centering of the responses and predictors before fitting
NOCVSTDIZE	Suppresses re-centering and rescaling of the responses and predictors when cross-validating
NOSCALE	Suppresses scaling of the responses and predictors before fitting
Output Options	
CENSCALE	Displays the centering and scaling information
DETAILS	Displays the details of the fitted model
NOCLPRINT	Limits or suppresses the display of class levels

Table 59.1 *continued*

Option	**Description**
NOPRINT	Suppresses ODS output
VARSS	Displays the amount of variation accounted for in each response and predictor

The following list provides details about these *options*.

CENSCALE

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

CVTEST < (*cvtest-options*) >

requests that van der Voet's (1994) randomization-based model comparison test be performed to test models that have different numbers of extracted factors against the model that minimizes the predicted residual sum of squares. For more information, see the section "Test Set Validation" on page 4609. You can also specify the following *cvtest-options* in parentheses after the CVTEST option:

PVAL=*n*

specifies the cutoff probability for declaring an insignificant difference. By default, PVAL=0.10.

STAT=PRESS | T2

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. By default, STAT=T2.

NSAMP=*n*

specifies the number of randomizations to perform. By default, NSAMP=1000.

SEED=*n*

specifies the seed value for the random number stream. If you do not specify a seed, or if you specify a value less than or equal to 0, the seed is generated from reading the time of day from the computer's clock.

Analyses that use the same (nonzero) seed are not completely reproducible if they are executed on a different number of threads because the random number streams in separate threads are independent. You can control the number of threads on which the HPPLS procedure executes by using SAS system options or by using the PERFORMANCE statement in the HPPLS procedure.

DATA=*SAS-data-set*

names the input SAS data set to be used by PROC HPPLS. The default is the most recently created data set.

If PROC HPPLS executes in distributed mode, the input data are distributed to memory on the appliance nodes and analyzed in parallel, unless the data are already distributed in the appliance database. In that case PROC HPPLS reads the data alongside the distributed database. For more information about the various execution modes, see the section "Processing Modes" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*). For more information about the alongside-the-database model, see the section "Alongside-the-Database Execution" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

DETAILS

lists the details of the fitted model for each successive factor. The listed details are different for different extraction methods. For more information, see the section "Displayed Output" on page 4612.

METHOD=PLS< (*PLS-options*) > | SIMPLS | PCR | RRR

specifies the general factor extraction method to be used. You can specify the following values:

PCR

requests principal components regression.

PLS< (*PLS-options*) >

requests partial least squares. You can also specify the following optional *PLS-options* in parentheses after METHOD=PLS:

ALGORITHM=NIPALS | SVD | EIG

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 $\mathbf{X}'\mathbf{Y}$, and EIG bases the extraction on the eigenvalue decomposition of $\mathbf{Y}'\mathbf{X}\mathbf{X}'\mathbf{Y}$. ALGORITHM=SVD is the most accurate but least efficient approach. By default, ALGORITHM=NIPALS.

EPSILON=*n*

specifies the convergence criterion for the NIPALS algorithm. By default, EPSILON=10^{-12}.

MAXITER=*n*

specifies the maximum number of iterations for the NIPALS algorithm. By default, MAXITER=200.

RRR

requests reduced rank regression.

SIMPLS

requests the straightforward implementation of a statistically inspired modification of the partial least squares (SIMPLS) method of De Jong (1993).

By default, METHOD=PLS.

NAMELEN=*number*

specifies the length to which long effect names are shortened. By default, NAMELEN=20. If you specify a value less than 20 for *number*, the default is used.

NFAC=*n*

specifies the number of factors to extract. The default is $\min\{15, p, N\}$, where p is the number of predictors (or the number of dependent variables when METHOD=RRR) and N is the number of runs (observations). You probably do not need to extract this many factors 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, you should also either specify the PARTITION statement 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 option is useful if the analysis variables are already centered and scaled. For more information, see the section "Centering and Scaling" on page 4610.

NOCLPRINT< **=***number* **>**

suppresses the display of the "Class Level Information" table if you do not specify *number*. If you specify *number*, the values of the classification variables are displayed only for variables whose number of levels is less than *number*. Specifying a *number* helps to reduce the size of the "Class Level Information" table if some classification variables have a large number of levels.

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 4610.

NOPRINT

suppresses the normal display of results. This option is useful when you want only the output statistics saved in a data set. 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 option is useful if the analysis variables are already centered and scaled. For more information, see the section "Centering and Scaling" on page 4610.

VARSS

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

BY *variables* **;**

You can specify a BY statement with PROC HPPLS 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 HPPLS 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 *SAS Visual Data Management and Utility Procedures Guide*.

CLASS Statement

CLASS *variable* < **(***options***)** >. . . < *variable* < **(***options***)** > > < / *global-options* > **;**

The CLASS statement names the classification variables to be used as explanatory variables in the analysis. The CLASS statement must precede the MODEL statement.

The CLASS statement for SAS high-performance statistical procedures is documented in the section "CLASS Statement" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*). The HPPLS procedure also supports the following *global-option* in the CLASS statement:

UPCASE

uppercases the values of character-valued CLASS variables before levelizing them. For example, if the UPCASE option is in effect and a CLASS variable can take the values 'a', 'A', and 'b', then 'a' and 'A' represent the same level and that CLASS variable is treated as having only two values: 'A' and 'B'.

ID Statement

ID *variables* **;**

The ID statement lists one or more variables from the input data set to be transferred to output data sets that are created by SAS high-performance analytical procedures, provided that the output data set produces one (or more) records per input observation.

For information about the common ID statement in SAS high-performance analytical procedures, see the section "ID Statement" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

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. You can simply list the names of the predictor variables as the model effects, but you can also specify other types of effects, including polynomial effects and interactions. For information about constructing the model effects, see the section "Specification and Parameterization of Model Effects" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

The MODEL statement is required. You can specify only one MODEL statement.

You can specify the following *options* in the MODEL statement after a slash (/).

INTERCEPT

overrides the default, in which the responses and predictors are centered. When responses and predictors are centered, no intercept is required in the model.

SOLUTION

lists the coefficients of the final predictive model for the responses. The coefficients for predicting the centered and scaled responses that are based on the centered and scaled predictors are displayed, in addition to the coefficients for predicting the raw responses that are based on the raw predictors.

OUTPUT Statement

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

The OUTPUT statement creates a data set that contains observationwise statistics, which are computed after fitting the model. If you do not specify any *keyword*, then only the predicted values for responses are included.

The variables in the input data set are *not* included in the output data set in order to avoid data duplication for large data sets; however, variables specified in the ID statement are included. If the input data are in distributed form, where accessing data in a particular order cannot be guaranteed, the HPPLS procedure copies the distribution or partition key to the output data set so that its contents can be joined with the input data.

You can specify the following syntax elements in the OUTPUT statement:

OUT=*SAS-data-set*

DATA=*SAS-data-set*

specifies the name of the output data set. If the OUT= (or DATA=) option is omitted, the procedure uses the DATA*n* convention to name the output data set.

keyword **<** **=***prefix* **>**

specifies a statistic to include in the output data set and optionally a *prefix* for naming the output variables. If you do not provide a *prefix*, the HPPLS procedure assigns a default prefix based on the type of statistic requested. For example, for response variables y1 and y2, a specification of PREDICTED produces two predicted value variables Pred_y1 and Pred_y2.

You can specify the following *keywords* for adding statistics to the OUTPUT data set:

H

requests the approximate leverage. The default prefix is H.

PREDICTED

PRED

P

requests predicted values for each response. The default prefix is Pred.

PRESS

requests approximate predicted residuals for each response. The default prefix is PRESS.

ROLE

requests numeric values that indicate the role played by each observation in fitting the model. The default prefix is _ROLE_. Table 59.2 shows the interpretation of this variable for each observation.

Table 59.2 Role Interpretation

Value	Observation Role
0	Not used
1	Training
2	Testing

If you do not partition the input data by using a PARTITION statement, then the role variable value is 1 for observations that are used in fitting the model, and 0 for observations that have at least one missing or invalid value for the responses or predictors.

STDX

requests standardized (centered and scaled) predictor values for each predictor. The default prefix is StdX.

STDXSSE

requests the sum of squares of residuals for standardized predictors. The default prefix is StdXSSE.

STDY

requests standardized (centered and scaled) response values for each response. The default prefix is StdY.

STDYSSE

requests the sum of squares of residuals for standardized responses. The default prefix is StdYSSE.

TSQUARE

T2

requests scaled sum of squares of score values. The default prefix is TSquare.

XRESIDUAL

XRESID

XR

requests residuals for each predictor. The default prefix is XResid.

XSCORE

requests extracted factors (X-scores, latent vectors, latent variables, *T*) for each selected model factor. The default prefix is XScore.

YRESIDUAL

YRESID

YR

requests residuals for each response. The default prefix is YResid.

YSCORE

requests extracted responses (Y-scores, U) for each selected model factor. The default prefix is YScore.

According to the *keyword* specified, the output variables that contain the requested statistic are named as follows:

- The *keywords* XRESIDUAL and STDX define an output variable for each predictor, so the variables that correspond to each predictor are named by appending a number (which starts from 1) to the prefix. For each defined variable, a label is also generated automatically; the label contains the prefix of the variable and the name of the predictor. For example, if the model has three predictors, then a specification of XRESIDUAL=XR produces the variables XR1, XR2, and XR3.
- The *keywords* PREDICTED, YRESIDUAL, STDY, and PRESS define an output variable for each response, so the variables that correspond to each response are named by appending the name of the response variable to the prefix. For example, if the model has response variables y1 and y2, then a specification of PREDICTED=P produces the variables P_y1 and P_y2.
- The *keywords* XSCORE and YSCORE define an output variable for each selected model factor, so the variables that correspond to each successive factor are named by appending the factor number to the prefix. For example, if the model has three selected factors, then a specification of XSCORE=T produces the variables T1, T2, and T3.
- The *keywords* H, TSQUARE, STDXSSE, STDYSSE, and ROLE each define a single output variable, so the variable name matches the prefix.

PARTITION Statement

PARTITION < *partition-options* > **;**

The PARTITION statement specifies how observations in the input data set are logically partitioned into disjoint subsets for model training and testing. Either you can designate a variable in the input data set and a set of formatted values of that variable to determine the role of each observation, or you can specify proportions to use for random assignment of observations for each role.

You can specify one (but not both) of the following *partition-options*:

ROLEVAR | ROLE=*variable* **(< TEST='***value***' > < TRAIN='***value***' >)**

names the variable in the input data set whose values are used to assign roles to each observation. The formatted values of this variable that are used to assign observations roles are specified in the TEST= and TRAIN= suboptions. If you specify only the TEST= suboption, then all observations whose role is not determined by the TEST= suboption are assigned to training. If you specify only the TRAIN= suboption, then all observations whose role is not determined by the TRAIN= suboption are assigned to testing.

FRACTION(<TEST=*fraction*> <SEED=*n*>)

requests that specified proportions of the observations in the input data set be randomly assigned training and testing roles. You specify the proportions for testing by using the TEST= suboption; the specified fraction must be less than 1 and the remaining fraction of the observations are assigned to the training role. If you do not specify the TEST= suboption, the default fraction is 0.5. The SEED= suboption specifies an integer that is used to start the pseudorandom number generator for random partitioning of data for training and testing. If you do not specify a seed, or if you specify a value less than or equal to 0, the seed is generated from reading the time of day from the computer's clock.

Because *fraction* is a per-observation probability (which means that any particular observation has a probability of *fraction* of being assigned the testing role), using the FRACTION option can cause different numbers of observations to be assigned training and testing roles. Different partitions can be observed when the number of nodes or threads changes or when PROC HPPLS runs in alongside-the-database mode.

PERFORMANCE Statement

PERFORMANCE <*performance-options*> **;**

The PERFORMANCE statement defines performance parameters for multithreaded and distributed computing, passes variables that describe the distributed computing environment, and requests detailed results about the performance characteristics of the HPPLS procedure.

You can also use the PERFORMANCE statement to control whether the HPPLS procedure executes in single-machine mode or distributed mode.

The PERFORMANCE statement is documented further in the section "PERFORMANCE Statement" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

Details: HPPLS Procedure

Regression Methods

All the predictive methods that PROC HPPLS implements 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 $\mathbf{X} = \mathbf{X}_0$ be the centered and scaled matrix of predictors, and let $\mathbf{Y} = \mathbf{Y}_0$ be the centered and scaled matrix of response values. The PLS method starts with a linear combination $\mathbf{t} = \mathbf{X}_0\mathbf{w}$ of the predictors, where $\mathbf{t}$ is called a *score vector* and $\mathbf{w}$ is its associated *weight vector*. The PLS method predicts both $\mathbf{X}_0$ and $\mathbf{Y}_0$ by regression on $\mathbf{t}$:

$$\begin{array}{rclcrcl} \hat{\mathbf{X}}_0 &=& \mathbf{t}\mathbf{p}', & \text{where} & \mathbf{p}' &=& (\mathbf{t}'\mathbf{t})^{-1}\mathbf{t}'\mathbf{X}_0 \\ \hat{\mathbf{Y}}_0 &=& \mathbf{t}\mathbf{c}', & \text{where} & \mathbf{c}' &=& (\mathbf{t}'\mathbf{t})^{-1}\mathbf{t}'\mathbf{Y}_0 \end{array}$$

The vectors $\mathbf{p}$ and $\mathbf{c}$ are called the X- and Y-*loadings*, respectively.

The specific linear combination $\mathbf{t} = \mathbf{X}_0\mathbf{w}$ is the one that has maximum covariance $\mathbf{t}'\mathbf{u}$ with some response linear combination $\mathbf{u} = \mathbf{Y}_0\mathbf{q}$. Another characterization is that the X-weight, $\mathbf{w}$, and the Y-weight, $\mathbf{q}$, are proportional to the first left and right singular vectors, respectively, of the covariance matrix $\mathbf{X}_0'\mathbf{Y}_0$ or, equivalently, the first eigenvectors of $\mathbf{X}_0'\mathbf{Y}_0\mathbf{Y}_0'\mathbf{X}_0$ and $\mathbf{Y}_0'\mathbf{X}_0\mathbf{X}_0'\mathbf{Y}_0$, respectively.

This accounts for how the first PLS factor is extracted. The second factor is extracted in the same way by replacing $\mathbf{X}_0$ and $\mathbf{Y}_0$ with the X- and Y-residuals from the first factor:

$$\begin{aligned}\mathbf{X}_1 &= \mathbf{X}_0 - \hat{\mathbf{X}}_0\\ \mathbf{Y}_1 &= \mathbf{Y}_0 - \hat{\mathbf{Y}}_0\end{aligned}$$

These residuals are also called the *deflated* $\mathbf{X}$ and $\mathbf{Y}$ 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 $\mathbf{X}_i$. This leads to difficulties in comparing different scores, weights, and so on. The SIMPLS method of De Jong (1993) overcomes these difficulties by computing each score $\mathbf{t}_i = \mathbf{X}\mathbf{r}_i$ in terms of the original (centered and scaled) predictors $\mathbf{X}$. The SIMPLS X-weight vectors r_i are similar to the eigenvectors of $\mathbf{S}\mathbf{S}' = \mathbf{X}'\mathbf{Y}\mathbf{Y}'\mathbf{X}$, but they satisfy a different orthogonality condition. The $\mathbf{r}_1$ vector is just the first eigenvector $\mathbf{e}_1$ (so that the first SIMPLS score is the same as the first PLS score). However, the second eigenvector maximizes

$$\mathbf{e}_1'\mathbf{S}\mathbf{S}'\mathbf{e}_2 \text{ subject to } \mathbf{e}_1'\mathbf{e}_2 = 0$$

whereas the second SIMPLS weight $\mathbf{r}_2$ maximizes

$$\mathbf{r}_1' S S' \mathbf{r}_2 \text{ subject to } \mathbf{r}_1'\mathbf{X}'\mathbf{X}\mathbf{r}_2 = \mathbf{t}_1'\mathbf{t}_2 = 0$$

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 because the scores are all defined in terms of $\mathbf{X}$, it is easy to compute the overall model coefficients $\mathbf{B}$:

$$\begin{aligned}\hat{\mathbf{Y}} &= \sum_i \mathbf{t}_i\mathbf{c}_i'\\ &= \sum_i \mathbf{X}\mathbf{r}_i\mathbf{c}_i'\\ &= \mathbf{X}\mathbf{B}, \text{ where } \mathbf{B} = \mathbf{R}\mathbf{C}'\end{aligned}$$

Principal Components Regression

Like the SIMPLS method, principal component regression (PCR) defines all the scores in terms of the original (centered and scaled) predictors $\mathbf{X}$. However, unlike both the PLS and SIMPLS methods, the PCR method chooses the X-weights and X-scores without regard to the response data. The X-scores are chosen to explain as much variation in $\mathbf{X}$ as possible; equivalently, the X-weights for the PCR method are the eigenvectors of the predictor covariance matrix $\mathbf{X}'\mathbf{X}$. 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 $\mathbf{Y}$ in terms of the original predictors $\mathbf{X}$.

Reduced Rank Regression

As discussed in the preceding sections, partial least squares depends on selecting factors $\mathbf{t} = \mathbf{X}\mathbf{w}$ of the predictors and $\mathbf{u} = \mathbf{Y}\mathbf{q}$ of the responses that have maximum covariance, whereas principal components regression effectively ignores $\mathbf{u}$ and selects $\mathbf{t}$ to have maximum variance, subject to orthogonality constraints. In contrast, reduced rank regression selects $\mathbf{u}$ 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 $\mathbf{q}_i$ are the eigenvectors of the covariance matrix $\hat{\mathbf{Y}}'_{\mathrm{LS}}\hat{\mathbf{Y}}_{\mathrm{LS}}$ of the responses that are predicted by ordinary least squares regression, and the X-scores are the projections of the Y-scores $\mathbf{Y}\mathbf{q}_i$ 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, because the sampling affects how well the model can extrapolate to future observations. All the techniques that the HPPLS procedure implements work by extracting successive factors (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 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, which consists of two predictors and one response:

```
data data;
   input x1 x2 y;
   datalines;
    3.37651  2.30716        0.75615
    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 hppls data=data nfac=1 method=rrr;
   model y = x1 x2;
run;

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

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

The amount of model and response variation that are explained by the first factor for each method is shown in Figure 59.11 through Figure 59.13.

Figure 59.11 Variation Explained by the First Reduced Rank Regression Factor

The HPPLS Procedure

Percent Variation Accounted for by Reduced Rank Regression Factors

	Model Effects		Dependent Variables	
Number of Extracted Factors	Current	Total	Current	Total
1	15.06605	15.06605	100.00000	100.00000

Figure 59.12 Variation Explained by the First Principal Components Regression Factor

The HPPLS Procedure

Percent Variation Accounted for by Principal Components

	Model Effects		Dependent Variables	
Number of Extracted Factors	Current	Total	Current	Total
1	92.99959	92.99959	9.37874	9.37874

Figure 59.13 Variation Explained by the First Partial Least Squares Regression Factor

The HPPLS Procedure

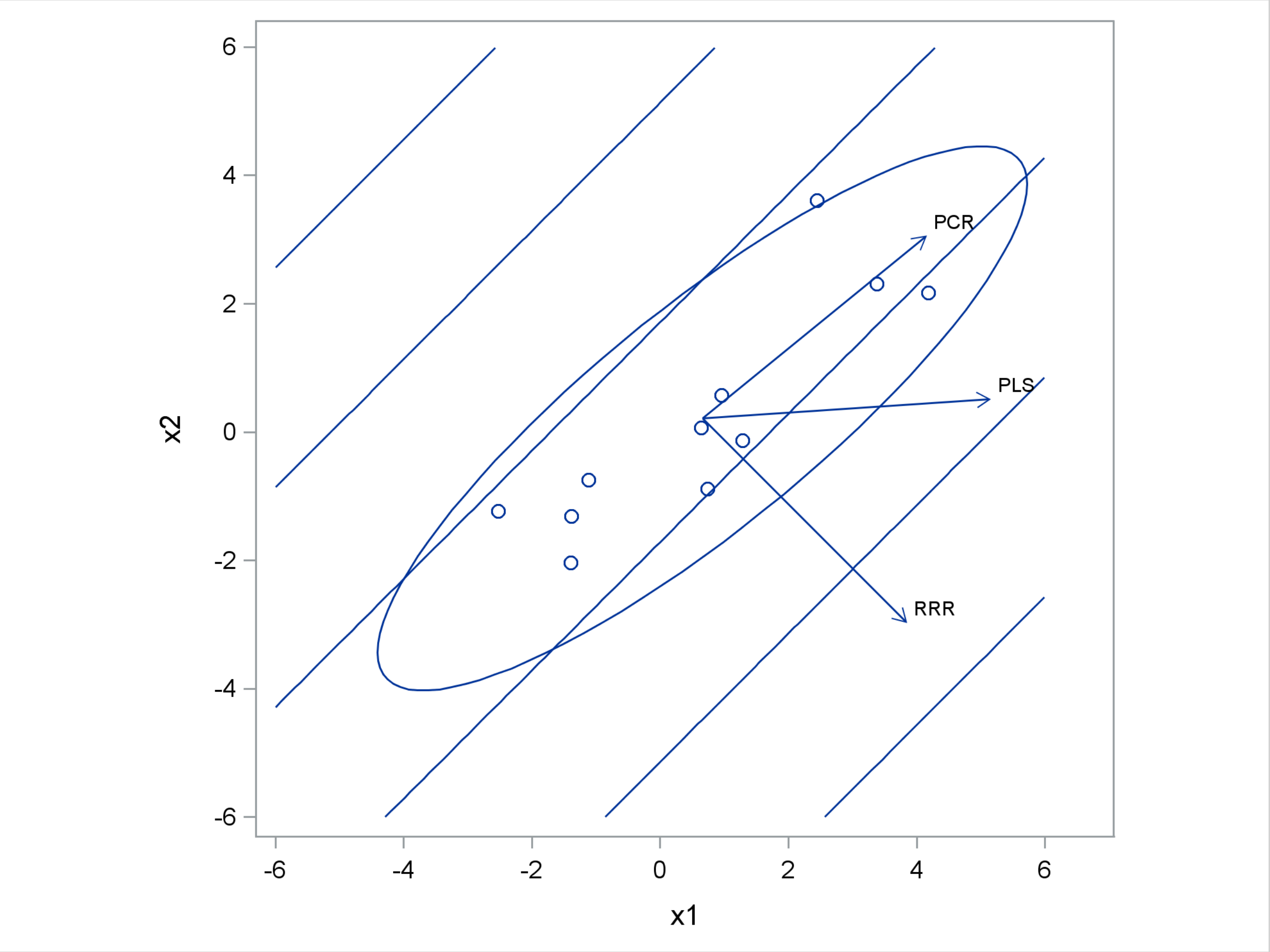
Percent Variation Accounted for by Partial Least Squares Factors

	Model Effects		Dependent Variables	
Number of Extracted Factors	Current	Total	Current	Total
1	88.53567	88.53567	26.53038	26.53038

Notice that although 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 component 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 59.14 illustrates how partial least squares balances the goals of explaining response and predictor variation in this case.

Figure 59.14 Depiction of the First Factors for Three Different Regression Methods



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 although the predictors vary most in the x1 = x2 direction, the response changes most in the orthogonal x1 = –x2 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.

Test Set Validation

None of the regression methods that the HPPLS procedure implements fit the observed data any better than ordinary least squares (OLS) regression; in fact, all the methods approach OLS as more factors are extracted. 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. So the crucial point is that when there are many predictors, OLS can *overfit* the observed data; biased regression methods that use 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 that are not involved in the modeling procedure itself.

The method of choosing the number of extracted factors that PROC HPPLS implements is called *test set validation*. When you have sufficient data, you can subdivide your data into two parts: training data and test data. During the validation process, the model is fit on the training data, and the predicted residual sum of squares (PRESS) for models that have different numbers of extracted factors is found by using the test data. The number of factors chosen is usually the one that minimizes PRESS.

You use a PARTITION statement to logically subdivide the DATA= data set into separate roles. You can name the fractions of the data that you want to reserve as training data and test data. For example, the following statements randomly subdivide the inData data set, reserving 50% each for training and testing:

```
proc hppls data=inData;
   partition fraction(test=0.5);
   ...
run;
```

In some cases you might need to exercise more control over the partitioning of the input data set. You can do this by naming both a variable in the input data set and a formatted value of that variable for each role. For example, the following statements assign roles to the observations in the inData data set based on the value of the variable Group in that data set. Observations whose value of Group is `'group 1'` are assigned for training, and those whose value is `'group 2'` are assigned to testing. All other observations are ignored.

```
proc hppls data=inData;
   partition roleVar=Group(train='group 1' test='group 2')
   ...
run;
```

By default, the number of extracted factors is chosen to be the one that minimizes PRESS. However, models that have fewer factors often have PRESS statistics that are only marginally larger than the absolute minimum. To address this, Van der Voet (1994) 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 while still producing residuals that are insignificantly larger than the residuals of the model that has a minimum PRESS.

To see how van der Voet's test works, let $R_{i,jk}$ be the jth predicted residual for response k for the model that has i extracted factors. Then, the PRESS statistic is $\sum_{jk} R_{i,jk}^2$. Also, let $i_{\min}$ 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:

$$D_{i,jk} \quad = \quad R_{i,jk}^2 - R_{i_{\min},jk}^2$$

One alternative for the critical value is $C_i = \sum_{jk} D_{i,jk}$, which is simply the difference between the PRESS statistics for i and $i_{\min}$ factors; alternatively, van der Voet suggests Hotelling's T^2 statistic $C_i = \mathbf{d}'_{i,\cdot}\mathbf{S}_i^{-1}\mathbf{d}_{i,\cdot}$, where $\mathbf{d}_{i,\cdot}$ 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 $R^2_{i,jk}$ and $R^2_{i_{\min},jk}$. 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 that are chosen is the least number of factors that have 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 in both. (For more information about how different methods explain variation, see the section "Regression Methods" on page 4604.) 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 std(Time) in Time is approximately equivalent to a change of std(Temp) 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 HPPLS 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.

If the predictors involve crossproduct terms, PROC HPPLS does not standardize the variables before it standardizes the crossproduct. That is, if the *i*th values of two predictors are denoted x_i^1 and x_i^2, then the default standardized *i*th value of the crossproduct is

$$\frac{x_i^1 x_i^2 - \mathrm{mean}_j(x_j^1 x_j^2)}{\mathrm{std}_j(x_j^1 x_j^2)}$$

When test set validation is performed for the number of effects, some practitioners disagree as to whether the training data should be retransformed. By default, PROC HPPLS does retransform the training data, but you can suppress this behavior by specifying the NOCVSTDIZE option in the PROC HPPLS statement.

Missing Values

Observations that have any missing independent variables (including all classification variables) are excluded from the analysis, and no predictions are computed for such observations. Observations that have no missing independent variables but do have missing dependent variables are also excluded from the analysis, but predictions are computed. If you use the PARTITION statement and specify the ROLEVAR= option, observations that contain the missing ROLEVAR= variable are excluded from the analysis, but predictions are computed for them.

Computational Method

Multithreading

Threading refers to the organization of computational work into multiple tasks (processing units that can be scheduled by the operating system). A task is associated with a thread. Multithreading refers to the concurrent execution of threads. When multithreading is possible, substantial performance gains can be realized compared to sequential (single-threaded) execution.

The number of threads that the HPPLS procedure spawns is determined by the number of CPUs on a machine and can be controlled in the following ways:

- You can specify the CPU count by using the CPUCOUNT= SAS system option. For example, if you specify the following statements, the HPPLS procedure schedules threads as if it executed on a system that has four CPUs, regardless of the actual CPU count.

  ```
  options cpucount=4;
  ```

- You can specify the NTHREADS= option in the PERFORMANCE statement to determine the number of threads. This specification overrides the system option. Specify NTHREADS=1 to force single-threaded execution.

The number of threads per machine is displayed in the “Performance Information” table, which is part of the default output. The HPPLS procedure allocates one thread per CPU.

The tasks that the HPPLS procedure multithreads are primarily defined by dividing the data that are processed on a single machine among the threads—that is, the HPPLS procedure implements multithreading through a data-parallel model. For example, if the input data set has 1,000 observations and PROC HPPLS runs on four threads, then 250 observations are associated with each thread. All operations that require access to the data are then multithreaded. These operations include the following:

- variable levelization
- effect levelization
- formation of the crossproducts matrix
- computation of loadings, weights, scores, generalized inverse, and residual sums of squares

- scoring of observations

In addition, operations on matrices such as sweeps might be multithreaded if the matrices are of sufficient size to realize performance benefits from managing multiple threads for the particular matrix operation.

Output Data Set

When an observationwise output data set is created, many procedures in SAS software add the variables from the input data set to the output data set. High-performance statistical procedures assume that the input data sets can be large and contain many variables. For performance reasons, the output data set contains only the following:

- variables that are explicitly created by the statement
- variables that are listed in the ID statement
- distribution keys or hash keys that are transferred from the input data set

Including these variables and keys enables you to add output data set information that is necessary for subsequent SQL joins without copying the entire input data set to the output data set. For more information about output data sets that are produced when PROC HPPLS runs in distributed mode, see the section "Output Data Sets" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

Displayed Output

The following sections describe the output that PROC HPPLS produces. The output is organized into various tables, which are discussed in the order of their appearance.

Performance Information

The "Performance Information" table is produced by default. It displays information about the execution mode. For single-machine mode, the table displays the number of threads used. For distributed mode, the table displays the grid mode (symmetric or asymmetric), the number of compute nodes, and the number of threads per node.

Data Access Information

The "Data Access Information" table is produced by default. For the input and output data sets, it displays the libref and data set name, the engine used to access the data, the role (input or output) of the data set, and the path that data followed to reach the computation.

Centering and Scaling Information

If you specify the CENSCALE option in the PROC HPPLS statement, the HPPLS procedure produces "Model Effect Centering and Scaling" and "Dependent Variable Centering and Scaling" tables, which display the centering and scaling information for each response and predictor.

Model Information

The “Model Information” table displays basic information about the model, such as the input data set, the factor extraction method, the validation method, and the type of parameterization used for classification variables that are named in the CLASS statement. If you use the PARTITION statement, the table also displays the random number seed for partition, the validation testing criterion, the number of random permutations, and the random number seed for permutation, depending on whether you specify the FRACTION option in the PARTITION statement and the CVTEST option in the PROC HPPLS statement.

Number of Observations

The “Number of Observations” table displays the number of observations that are read from the input data set and the number of observations that are used in the analysis. If you use a PARTITION statement, the table also displays the number of observations that are used for each data role.

Class Level Information

The “Class Level Information” table lists the levels of every variable that is specified in the CLASS statement. You should check this information to make sure that the data are correct. You can adjust the order of the CLASS variable levels by specifying the ORDER= option in the CLASS statement. You can suppress the “Class Level Information” table completely or partially by specifying the NOCLPRINT= option in the PROC HPPLS statement.

If the classification variables are in the reference parameterization, the “Class Level Information” table also displays the reference value for each variable.

Dimensions

The “Dimensions” table displays information about the number of response variables, the number of effects, and the number of predictor parameters. It also displays the number of factors to extract.

Test Set Validation

If you use the PARTITION statement to perform a test set validation for choosing the number of extracted factors, the HPPLS procedure produces a “Test Set Validation Residual Summary” table to display a residual summary of the validation for each number of factors. It also produces a “Test Set Validation Results” table to display information about the optimal number of factors.

Percent Variation Accounted for by Extracted Factors

By default, the HPPLS procedure produces the “Percent Variation Accounted for by Extracted Factors” table to display just the amount of predictor variation and response variation that are accounted for by each factor. If you specify the VARSS option in the PROC HPPLS statement, the HPPLS procedure also produces the “Model Effect Percent Variation Accounted for by Extracted Factors” table and the “Dependent Variable Percent Variation Accounted for by Extracted Factors” table to display the amount of variation that is accounted for in each response and predictor, in addition to the average response and predictor sum of squares that are accounted for by each successive factor.

Model Details

If you specify the DETAILS option in the PROC HPPLS statement, the HPPLS procedure produces tables to display details about the fitted model for each successive factor. These tables include the following:

- “Model Effect Loadings” table, which displays the predictor loadings
- “Model Effect Weights” table, which displays predictor weights
- “Dependent Variable Weights” table, which displays the response weights
- “Coded Regression Coefficients” tables, which display the coded regression coefficients, if you specify METHOD=SIMPLS, METHOD=PCR, or METHOD=RRR in the PROC HPPLS statement.

Parameter Estimates

If you specify the SOLUTION option in the MODEL statement, the HPPLS procedure produces a “Parameter Estimates” table to display 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, in addition to the coefficients for predicting the raw responses based on the raw predictors.

Timing Information

If you specify the DETAILS option in the PERFORMANCE statement, the HPPLS procedure produces a “Timing” table, which displays the elapsed time (absolute and relative) of each main task of the procedure.

ODS Table Names

PROC HPPLS assigns a name to each table that it creates. You can use these names to refer to the ODS table when you use the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 59.3. For more information about ODS, see Chapter 20, “Using the Output Delivery System.”

Table 59.3 ODS Tables Produced by PROC HPPLS

Table Name	Description	Required Statement and Option
CVResults	Results of test set validation	PARTITION statement
CenScaleParms	Parameter estimates for centered and scaled data	SOLUTION option in MODEL statement
ClassLevels	Level information from the CLASS statement	CLASS statement
CodedCoef	Coded regression coefficients	DETAILS option in PROC HPPLS statement
DataAccessInfo	Information about modes of data access	Default output
Dimensions	Model dimensions	Default output
ModelInfo	Model information	Default output
NObs	Number of observations read and used	Default output

Table 59.3 *continued*

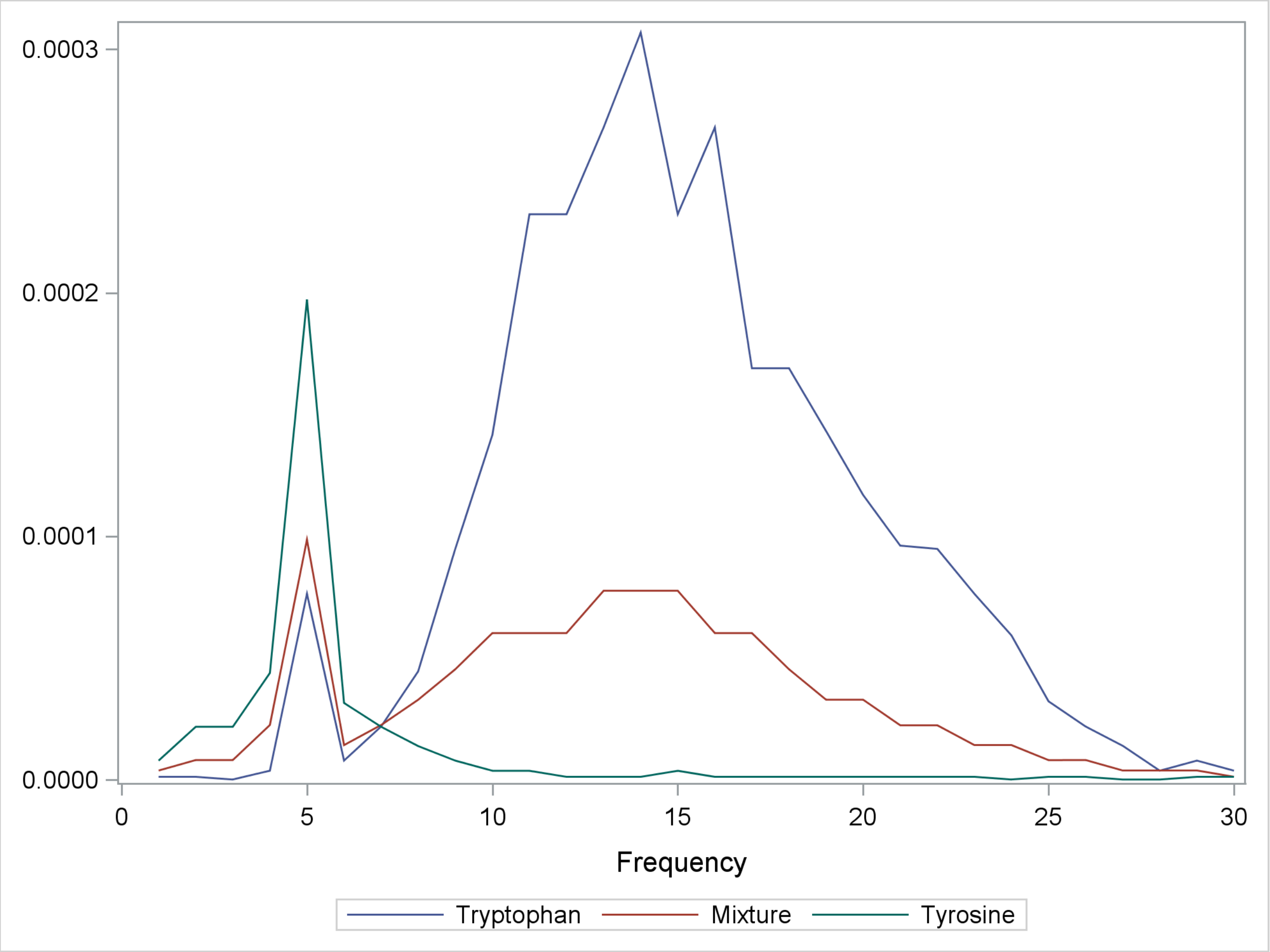
Table Name	Description	Required Statement and Option
ParameterEstimates	Parameter estimates for raw data	SOLUTION option in MODEL statement
PercentVariation	Predictor and response variation that are accounted for by each factor	Default output
PerformanceInfo	Information about the high-performance computing environment	Default output
ResidualSummary	Residual summary from test set validation	PARTITION statement
Timing	Absolute and relative times of tasks that are performed by the procedure	DETAILS option in PERFORMANCE statement
XEffectCenScale	Centering and scaling information for predictor effects	CENSCALE option in PROC HPPLS statement
XLoadings	Loadings for predictor effects	DETAILS option in PROC HPPLS statement
XPercentVariation	Variation that is accounted for by each factor for predictor effects	VARSS option in PROC HPPLS statement
XWeights	Weights for predictor effects	DETAILS option in PROC HPPLS statement
YPercentVariation	Variation that is accounted for by each factor for responses	VARSS option in PROC HPPLS statement
YVariableCenScale	Centering and scaling information for responses	CENSCALE option in PROC HPPLS statement
YWeights	Weights for responses	DETAILS option in PROC HPPLS statement

Examples: HPPLS Procedure

Example 59.1: 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 that contain known concentrations of two amino acids, tyrosine and tryptophan. The spectra are measured at 30 frequencies across the overall range of frequencies. For example, Output 59.1.1 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}.

Output 59.1.1 Spectra for Three Samples of Tyrosine and Tryptophan



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, where the concentrations are 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, in addition to the logarithm of the total concentration. Actually, because zeros are possible in both the responses and the predictors, slightly different transformations are used. The following statements create a SAS data set named ex1Data for these data. The data set also contains a variable Role that is used to assign samples to the training and testing roles.

```
data ex1Data;
   input obsnam $ Role : $5. 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 TRAIN 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
 -1.520 -1.804 -2.044 -2.269 -2.496 -2.714
19mix35 TRAIN 0.00003 3E-7
 -5.516 -5.294 -4.823 -3.858 -2.827 -2.249 -1.683 -1.218
 -0.907 -0.658 -0.501 -0.400 -0.345 -0.323 -0.342 -0.387
 -0.461 -0.554 -0.665 -0.803 -0.887 -0.960 -1.072 -1.272
 -1.541 -1.814 -2.058 -2.289 -2.496 -2.712
21mix35 TRAIN 0.00003 7.5E-7
 -5.519 -5.294 -4.501 -3.863 -2.827 -2.280 -1.716 -1.262
 -0.939 -0.694 -0.536 -0.444 -0.384 -0.369 -0.377 -0.421
 -0.495 -0.596 -0.706 -0.824 -0.917 -0.988 -1.103 -1.294
 -1.565 -1.841 -2.084 -2.320 -2.521 -2.729
23mix35 TRAIN 0.00003 1.5E-6

   ... more lines ...

26tyro5 TEST 0.00001 0.00001
 -3.037 -2.696 -2.464 -2.321 -2.239 -2.444 -2.602 -2.823
 -3.144 -3.396 -3.742 -4.063 -4.398 -4.699 -4.893 -5.138
 -5.140 -5.461 -5.463 -5.945 -5.461 -5.138 -5.140 -5.138
 -5.138 -5.463 -5.461 -5.461 -5.461 -5.461
tyro2   TEST 0.0001 0.0001
 -1.081 -0.710 -0.470 -0.337 -0.327 -0.433 -0.602 -0.841
 -1.119 -1.423 -1.750 -2.121 -2.449 -2.818 -3.110 -3.467
 -3.781 -4.029 -4.241 -4.366 -4.501 -4.366 -4.501 -4.501
 -4.668 -4.668 -4.865 -4.865 -5.109 -5.111
;
```

The following statements fit a PLS model that has 10 factors.

```
proc hppls data=ex1Data nfac=10;
   model tot_log tyr_log try_log = f1-f30;
run;
```

The “Model Information” table in Output 59.1.2 shows that no validation method is used. The “Number of Observations” table confirms that all 33 sample observations are used in the analysis.

The table in Output 59.1.3 indicates that only four or five factors are required to explain almost all of the variation in both the predictors and the responses.

Output 59.1.2 Model Information and Number of Observations

The HPPLS Procedure

Model Information	
Data Source	WORK.EX1DATA
Factor Extraction Method	Partial Least Squares
PLS Algorithm	NIPALS
Validation Method	None

Number of Observations Read	33
Number of Observations Used	33

Output 59.1.3 Amount of Variation Explained

Percent Variation Accounted for by Partial Least Squares Factors

	Model Effects		Dependent Variables	
Number of Extracted Factors	**Current**	**Total**	**Current**	**Total**
1	77.67903	77.67903	47.80217	47.80217
2	20.62719	98.30622	38.96826	86.77043
3	1.00143	99.30766	6.89262	93.66305
4	0.24930	99.55696	1.77222	95.43528
5	0.13077	99.68773	1.71762	97.15290
6	0.08970	99.77742	0.58619	97.73909
7	0.05684	99.83426	0.29079	98.02988
8	0.06730	99.90156	0.13857	98.16845
9	0.01521	99.91676	0.68214	98.85059
10	0.02627	99.94304	0.14388	98.99447

In order to choose the optimal number of PLS factors, you can explore how well models that are based on data in training roles and have different numbers of factors fit the data in testing roles. To do so, you can use the PARTITION statement to assign observations to training and testing roles based on the values of the input variable named Role.

```
proc hppls data=ex1Data nfac=10 cvtest(stat=press seed=12345);
   model tot_log tyr_log try_log = f1-f30;
   partition roleVar = Role(train='TRAIN' test='TEST');
run;
```

Output 59.1.4 shows the “Model Information” table and the “Number of Observations” table. The “Model Information” table indicates that test set validation is used and displays information about the options that are used in the model comparison test. The “Number of Observations” table confirms that there are 18 observations for the training role and 15 for the testing role.

Output 59.1.5 displays the results of the test set validation. They indicate that although five PLS factors produce the minimum predicted residual sum of squares, the residuals for four factors are insignificantly different from the residuals for five factors. Thus, the smaller model is preferred.

Output 59.1.4 Model Information and Number of Observations with Test Set Validation

The HPPLS Procedure

Model Information	
Data Source	WORK.EX1DATA
Factor Extraction Method	Partial Least Squares
PLS Algorithm	NIPALS
Validation Method	Test Set Validation
Validation Testing Criterion	Prob PRESS > 0.1
Number of Random Permutations	1000
Random Number Seed for Permutation	12345

Output 59.1.4 *continued*

Number of Observations Read	33
Number of Observations Used	33
Number of Observations Used for Training	18
Number of Observations Used for Testing	15

Output 59.1.5 Test Set Validation for the Number of PLS Factors

The HPPLS Procedure

Test Set Validation for the Number of Extracted Factors

Number of Extracted Factors	Root Mean PRESS	Prob > PRESS
0	3.056797	<.0001
1	2.630561	<.0001
2	1.00706	0.0070
3	0.664603	<.0001
4	0.521578	0.3760
5	0.500034	1.0000
6	0.513561	0.5000
7	0.501431	0.6850
8	1.055791	0.1520
9	1.435085	0.1010
10	1.720389	0.0330

Minimum Root Mean PRESS	0.500034
Minimizing Number of Factors	5
Smallest Number of Factors with p > 0.1	4

Percent Variation Accounted for by Partial Least Squares Factors

	Model Effects		Dependent Variables	
Number of Extracted Factors	**Current**	**Total**	**Current**	**Total**
1	81.16545	81.16545	48.33854	48.33854
2	16.81131	97.97676	32.54654	80.88508
3	1.76391	99.74067	11.44380	92.32888
4	0.19507	99.93574	3.83631	96.16519

Example 59.2: Fitting a PLS Model in Single-Machine and Distributed Modes

This example shows how you can run PROC HPPLS in single-machine and distributed modes. For more information about the execution modes of SAS high-performance analytical procedures, see the section "Processing Modes" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*). The focus of this example is to show how you can switch the modes of execution in PROC HPPLS. The following DATA step generates the data:

```
data ex2Data;
   drop i j k sign n n1 n2 n3 n4;

   n  = 100000;
   n1 = n*0.1;
   n2 = n*0.25;
   n3 = n*0.45;
   n4 = n*0.7;

   array y{10};
   array x{100};

   do i=1 to n;
      do j=1 to dim(y);
         y{j} = 1;
      end;
      sign = 1;

      do j=1 to dim(x);
         x{j} = ranuni(1);
         do k=1 to dim(y);
            y{k} = y{k} + sign*j*x{j};
            sign = -sign;
         end;
      end;

      do j=1 to dim(y);
         y{j} = y{j} + 7*rannor(1);
      end;

      if      i <= n1 then z='verytiny';
      else if i <= n2 then z='small';
      else if i <= n3 then z='medium';
      else if i <= n4 then z='large';
      else                 z='huge';

      output;
   end;
run;
```

The following statements use PROC HPPLS to fit a PLS model by using the SIMPLS method and test set validation:

```
proc hppls data=ex2Data method=simpls cvtest(stat=press seed=12345);
   class z;
   model y: = x: z:;
   partition fraction(test=0.4 seed=67890);
   performance details;
run;
```

In this example, any particular observation has a 40% probability of being assigned the testing role. All nonassigned observations are in training roles.

Output 59.2.1 shows the “Performance Information” table. This table shows that the HPPLS procedure executes in single-machine mode on four threads (the client machine has four CPUs). You can force a certain number of threads on any machine to be involved in the computations by specifying the NTHREADS= option in the PERFORMANCE statement.

Output 59.2.1 Performance Information in Single-Machine Mode

The HPPLS Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	4

Output 59.2.2 shows timing information for the PROC HPPLS run. This table is produced when you specify the DETAILS option in the PERFORMANCE statement. You can see that, in this case, the majority of time is spent fitting a PLS model.

Output 59.2.2 Timing in Single-Machine Mode

Procedure Task Timing		
Task	**Seconds**	**Percent**
Reading and Levelizing Data	0.71	0.64%
Fitting Model	110.53	99.36%

To switch to running PROC HPPLS in distributed mode, specify valid values for the NODES=, INSTALL=, and HOST= options in the PERFORMANCE statement. An alternative to specifying the INSTALL= and HOST= options in the PERFORMANCE statement is to use the OPTIONS SET commands to set appropriate values for the GRIDHOST and GRIDINSTALLLOC environment variables. For information about setting these options or environment variables, see the section “Processing Modes” (Chapter 2, *SAS/STAT User’s Guide: High-Performance Procedures*).

NOTE: Distributed mode requires SAS High-Performance Statistics.

The following statements provide an example. To run these statements successfully, you need to set the macro variables GRIDHOST and GRIDINSTALLLOC to resolve to appropriate values, or you can replace the references to macro variables with appropriate values.

```
proc hppls data=ex2Data method=simpls cvtest(stat=press seed=12345);
   class z;
   model y: = x: z:;
   partition fraction(test=0.4 seed=67890);
   performance details nodes = 4
               host="&GRIDHOST" install="&GRIDINSTALLLOC";
run;
```

The execution mode in the “Performance Information” table shown in Output 59.2.3 indicates that the calculations were performed in a distributed environment that uses four nodes, each of which uses 32 threads.

Output 59.2.3 Performance Information in Distributed Mode

Performance Information

Host Node	<< your grid host >>
Install Location	<< your grid install location >>
Execution Mode	Distributed
Number of Compute Nodes	4
Number of Threads per Node	32

Another indication of distributed execution is the following message, which is issued by all high-performance analytical procedures (with the corresponding procedure name) in the SAS log:

```
NOTE: The HPPLS procedure is executing in the distributed
      computing environment with 4 worker nodes.
```

Output 59.2.4 shows timing information for this distributed run of the HPPLS procedure. The majority of time in the distributed mode run is also spent fitting a model.

Output 59.2.4 Timing in Distributed Mode

Procedure Task Timing

Task	Seconds	Percent
Distributing Data	1.54	16.40%
Reading and Levelizing Data	0.37	3.91%
Fitting Model	7.29	77.73%
Waiting on Client	0.18	1.96%

References

De Jong, S. (1993). “SIMPLS: An Alternative Approach to Partial Least Squares Regression.” *Chemometrics and Intelligent Laboratory Systems* 18:251–263.

De Jong, S., and Kiers, H. (1992). “Principal Covariates Regression.” *Chemometrics and Intelligent Laboratory Systems* 14:155–164.

Dijkstra, T. K. (1983). “Some Comments on Maximum Likelihood and Partial Least Squares Methods.” *Journal of Econometrics* 22:67–90.

Dijkstra, T. K. (1985). *Latent Variables in Linear Stochastic Models: Reflections on Maximum Likelihood and Partial Least Squares Methods*. 2nd ed. Amsterdam: Sociometric Research Foundation.

Frank, I., and Friedman, J. (1993). “A Statistical View of Some Chemometrics Regression Tools.” *Technometrics* 35:109–135.

Geladi, P., and Kowalski, B. (1986). “Partial Least-Squares Regression: A Tutorial.” *Analytica Chimica Acta* 185:1–17.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. New York: Macmillan.

Helland, I. S. (1988). "On the Structure of Partial Least Squares Regression." *Communications in Statistics—Simulation and Computation* 17:581–607.

Hoerl, A., and Kennard, R. (1970). "Ridge Regression: Biased Estimation for Non-orthogonal Problems." *Technometrics* 12:55–67.

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

McAvoy, T. J., Wang, N. S., Naidu, S., Bhat, N., Gunter, J., and Simmons, M. (1989). "Interpreting Biosensor Data via Backpropagation." *International Joint Conference on Neural Networks* 1:227–233.

Naes, T., and Martens, H. (1985). "Comparison of Prediction Methods for Multicollinear Data." *Communications in Statistics—Simulation and Computation* 14:545–576.

Ränner, S., Lindgren, F., Geladi, P., and Wold, S. (1994). "A PLS Kernel Algorithm for Data Sets with Many Variables and Fewer Objects." *Journal of Chemometrics* 8:111–125.

Sarle, W. S. (1994). "Neural Networks and Statistical Models." In *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, 1538–1550. Cary, NC: SAS Institute Inc. `http://www.sascommunity.org/sugi/SUGI94/Sugi-94-255%20Sarle.pdf`.

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

Tobias, R. D. (1995). "An Introduction to Partial Least Squares Regression." In *Proceedings of the Twentieth Annual SAS Users Group International Conference*, 1250–1257. Cary, NC: SAS Institute Inc. `http://www.sascommunity.org/sugi/SUGI95/Sugi-95-210%20Tobias.pdf`.

Umetrics (1995). *Multivariate Analysis*. Three-day course. Winchester, MA: Umetrics.

Van den Wollenberg, A. L. (1977). "Redundancy Analysis: An Alternative to Canonical Correlation Analysis." *Psychometrika* 42:207–219.

Van der Voet, H. (1994). "Comparing the Predictive Accuracy of Models Using a Simple Randomization Test." *Chemometrics and Intelligent Laboratory Systems* 25:313–323.

Wold, H. (1966). "Estimation of Principal Components and Related Models by Iterative Least Squares." In *Multivariate Analysis*, edited by P. R. Krishnaiah, 391–420. New York: Academic Press.

Subject Index

Syntax Index