THE POWER TO KNOW.

SAS/STAT® 15.1 User's Guide

The HPPRINCOMP Procedure

This document is an individual chapter from *SAS/STAT® 15.1 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS/STAT® 15.1 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 15.1 User's Guide

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

November 2018

Chapter 62
The HPPRINCOMP Procedure

Contents

Overview: HPPRINCOMP Procedure

The HPPRINCOMP procedure is a high-performance procedure that performs principal component analysis. It is a high-performance version of the PRINCOMP procedure in SAS/STAT software, but it provides additional iterative methods to calculate the principal components.

Principal component analysis is a multivariate technique for examining relationships among several quantitative variables, providing an optimal way of reducing dimensionality by projecting the data onto a lower-dimensional orthogonal subspace that explains as much variation in those variables as possible. The choice between using factor analysis and using principal component analysis depends in part on your research objectives. You should use the HPPRINCOMP procedure if you are interested in summarizing data and detecting linear relationships. You can use principal component analysis to reduce the number of variables in regression, clustering, and so on.

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

NOTE: Distributed mode requires SAS High-Performance Statistics.

PROC HPPRINCOMP Features

The main features of the HPPRINCOMP procedure are as follows:

- supports a PARTIAL statement for analyzing a partial correlation or covariance matrix
- supports a FREQ statement for grouped analysis
- supports a WEIGHT statement for weighted analysis

- produces an output data set that contains principal component scores and other observationwise statistics
- produces an output data set that contains means, standard deviations, number of observations, correlations or covariances, eigenvalues, and eigenvectors

The HPPRINCOMP procedure implements the following algorithms:

- eigenvalue decomposition, which uses the correlation or covariance of the data matrix and calculates all the principal components simultaneously
- nonlinear iterative partial least squares (NIPALS), which uses the data matrix and extracts the principal components successively
- the iterative method based on Gram-Schmidt orthogonalization (ITERGS) of Andrecut (2009), which uses the data matrix and extracts the principal components successively. The algorithm applies reorthogonalization correction to both the scores and the loadings at each iteration step.

Because the HPPRINCOMP 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 HPPRINCOMP Contrasted with PROC PRINCOMP

The HPPRINCOMP procedure and the PRINCOMP procedure in SAS/STAT have the following similarities and differences:

- All statements that are available in PROC PRINCOMP are supported by the HPPRINCOMP procedure.
- The HPPRINCOMP procedure supports the OUTPUT statement, which is not available in PROC PRINCOMP.
- The HPPRINCOMP procedure can specify various methods to be used for calculating the principal components by using the METHOD= option, which is not available in PROC PRINCOMP.
- PROC PRINCOMP can accept ordinary SAS data sets and other types of special SAS data sets as input. The HPPRINCOMP procedure can accept only ordinary SAS data sets (raw data) as input.
- The HPPRINCOMP procedure does not support the PLOTS option that is available in PROC PRINCOMP.
- The HPPRINCOMP procedure is specifically designed to operate in the high-performance distributed environment. By default, PROC HPPRINCOMP performs computations on multiple threads. The PRINCOMP procedure executes on a single thread.

Getting Started: HPPRINCOMP Procedure

The following data provide crime rates per 100,000 people in seven categories for each of the 50 US states in 1977:

```
title 'Crime Rates per 100,000 Population by State';

data Crime;
   input State $1-15 Murder Rape Robbery Assault
         Burglary Larceny Auto_Theft;
   datalines;
Alabama        14.2 25.2  96.8 278.3 1135.5 1881.9 280.7
Alaska         10.8 51.6  96.8 284.0 1331.7 3369.8 753.3
Arizona         9.5 34.2 138.2 312.3 2346.1 4467.4 439.5
Arkansas        8.8 27.6  83.2 203.4  972.6 1862.1 183.4
California     11.5 49.4 287.0 358.0 2139.4 3499.8 663.5
Colorado        6.3 42.0 170.7 292.9 1935.2 3903.2 477.1
Connecticut     4.2 16.8 129.5 131.8 1346.0 2620.7 593.2
Delaware        6.0 24.9 157.0 194.2 1682.6 3678.4 467.0
Florida        10.2 39.6 187.9 449.1 1859.9 3840.5 351.4
Georgia        11.7 31.1 140.5 256.5 1351.1 2170.2 297.9
Hawaii          7.2 25.5 128.0  64.1 1911.5 3920.4 489.4
Idaho           5.5 19.4  39.6 172.5 1050.8 2599.6 237.6
Illinois        9.9 21.8 211.3 209.0 1085.0 2828.5 528.6
Indiana         7.4 26.5 123.2 153.5 1086.2 2498.7 377.4
Iowa            2.3 10.6  41.2  89.8  812.5 2685.1 219.9
Kansas          6.6 22.0 100.7 180.5 1270.4 2739.3 244.3
Kentucky       10.1 19.1  81.1 123.3  872.2 1662.1 245.4
Louisiana      15.5 30.9 142.9 335.5 1165.5 2469.9 337.7
Maine           2.4 13.5  38.7 170.0 1253.1 2350.7 246.9
Maryland        8.0 34.8 292.1 358.9 1400.0 3177.7 428.5
Massachusetts   3.1 20.8 169.1 231.6 1532.2 2311.3 1140.1
Michigan        9.3 38.9 261.9 274.6 1522.7 3159.0 545.5
Minnesota       2.7 19.5  85.9  85.8 1134.7 2559.3 343.1
Mississippi    14.3 19.6  65.7 189.1  915.6 1239.9 144.4
Missouri        9.6 28.3 189.0 233.5 1318.3 2424.2 378.4
Montana         5.4 16.7  39.2 156.8  804.9 2773.2 309.2
Nebraska        3.9 18.1  64.7 112.7  760.0 2316.1 249.1
Nevada         15.8 49.1 323.1 355.0 2453.1 4212.6 559.2
New Hampshire   3.2 10.7  23.2  76.0 1041.7 2343.9 293.4
New Jersey      5.6 21.0 180.4 185.1 1435.8 2774.5 511.5
New Mexico      8.8 39.1 109.6 343.4 1418.7 3008.6 259.5
New York       10.7 29.4 472.6 319.1 1728.0 2782.0 745.8
North Carolina 10.6 17.0  61.3 318.3 1154.1 2037.8 192.1
North Dakota    0.9  9.0  13.3  43.8  446.1 1843.0 144.7
Ohio            7.8 27.3 190.5 181.1 1216.0 2696.8 400.4
Oklahoma        8.6 29.2  73.8 205.0 1288.2 2228.1 326.8
Oregon          4.9 39.9 124.1 286.9 1636.4 3506.1 388.9
Pennsylvania    5.6 19.0 130.3 128.0  877.5 1624.1 333.2
Rhode Island    3.6 10.5  86.5 201.0 1489.5 2844.1 791.4
South Carolina 11.9 33.0 105.9 485.3 1613.6 2342.4 245.1
South Dakota    2.0 13.5  17.9 155.7  570.5 1704.4 147.5
```

```
   Tennessee      10.1 29.7 145.8 203.9 1259.7 1776.5 314.0
   Texas          13.3 33.8 152.4 208.2 1603.1 2988.7 397.6
   Utah            3.5 20.3  68.8 147.3 1171.6 3004.6 334.5
   Vermont         1.4 15.9  30.8 101.2 1348.2 2201.0 265.2
   Virginia        9.0 23.3  92.1 165.7  986.2 2521.2 226.7
   Washington      4.3 39.6 106.2 224.8 1605.6 3386.9 360.3
   West Virginia   6.0 13.2  42.2     .  597.4 1341.7 163.3
   Wisconsin       2.8 12.9  52.2  63.7  846.9 2614.2 220.7
   Wyoming          .  21.9  39.7 173.9  811.6 2772.2 282.0
   ;
```

The following statements invoke the HPPRINCOMP procedure, which requests a principal component analysis of the data and produces Figure 62.1 through Figure 62.4:

```
   proc hpprincomp data=Crime;
   run;
```

Figure 62.1 displays the “Performance Information,” “Data Access Information,” “Model Information,” “Number of Observations,” “Number of Variables,” and “Simple Statistics” tables.

The “Performance Information” table shows the procedure executes in single-machine mode—that is, the data reside and the computation is performed on the machine where the SAS session executes. This run of the HPPRINCOMP procedure took place on a multicore machine with 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 principal component extraction method is eigenvalue decomposition, which is the default.

The “Number of Observations” table shows that of the 50 observations in the input data, only 48 observations are used in the analysis because some observations have incomplete data.

The “Number of Variables” table indicates that there are seven variables to be analyzed and seven principal components to be computed. By default, if the VAR statement is omitted, all numeric variables that are not listed in other statements are used in the analysis.

The “Simple Statistics” table displays the mean and standard deviation of the analysis variables.

Figure 62.1 Performance Information and Simple Statistics

Crime Rates per 100,000 Population by State

The HPPRINCOMP Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

Data	Engine	Role	Path
WORK.CRIME	V9	Input	On Client

Figure 62.1 *continued*

Model Information	
Data Source	WORK.CRIME
Component Extraction Method	Eigenvalue Decomposition

Number of Observations Read	50
Number of Observations Used	48

Number of Variables	7
Number of Principal Components	7

Simple Statistics

Variable	Mean	Standard Deviation
Murder	7.51667	3.93059
Rape	26.07500	10.81304
Robbery	127.55625	88.49374
Assault	214.58750	100.64360
Burglary	1316.37917	423.31261
Larceny	2696.88542	714.75023
Auto_Theft	383.97917	194.37033

Figure 62.2 displays the "Correlation Matrix" table. By default, the PROC HPPRINCOMP statement requests that principal components be computed from the correlation matrix, so the total variance is equal to the number of variables, 7.

Figure 62.2 Correlation Matrix Table

Correlation Matrix

Variable	Murder	Rape	Robbery	Assault	Burglary	Larceny	Auto_Theft
Murder	1.0000	0.6000	0.4768	0.6485	0.3778	0.0925	0.0555
Rape	0.6000	1.0000	0.5817	0.7316	0.7038	0.6009	0.3282
Robbery	0.4768	0.5817	1.0000	0.5452	0.6200	0.4371	0.5787
Assault	0.6485	0.7316	0.5452	1.0000	0.6082	0.3791	0.2520
Burglary	0.3778	0.7038	0.6200	0.6082	1.0000	0.7932	0.5390
Larceny	0.0925	0.6009	0.4371	0.3791	0.7932	1.0000	0.4246
Auto_Theft	0.0555	0.3282	0.5787	0.2520	0.5390	0.4246	1.0000

Figure 62.3 displays the "Eigenvalues" table. The first principal component accounts for about 57.8% of the total variance, the second principal component accounts for about 18.1%, and the third principal component accounts for about 10.7%. Note that the eigenvalues sum to the total variance.

The eigenvalues indicate that two or three components provide a good summary of the data: two components account for 76% of the total variance, and three components account for 87%. Subsequent components account for less than 5% each.

Figure 62.3 Eigenvalues Table

Eigenvalues of the Correlation Matrix				
	Eigenvalue	Difference	Proportion	Cumulative
1	4.045824	2.781795	0.5780	0.5780
2	1.264030	0.516529	0.1806	0.7586
3	0.747500	0.421175	0.1068	0.8653
4	0.326325	0.061119	0.0466	0.9120
5	0.265207	0.036843	0.0379	0.9498
6	0.228364	0.105613	0.0326	0.9825
7	0.122750		0.0175	1.0000

Figure 62.4 displays the "Eigenvectors" table. From the eigenvectors matrix, you can represent the first principal component, Prin1, as a linear combination of the original variables:

$$\begin{aligned}\text{Prin1} = {} & 0.302888 \times (\text{Murder}) \\ & + 0.434103 \times (\text{Rape}) \\ & + 0.397055 \times (\text{Robbery}) \\ & . \\ & . \\ & . \\ & + 0.288343 \times (\text{Auto\_Theft})\end{aligned}$$

Similarly, the second principal component, Prin2, is

$$\begin{aligned}\text{Prin2} = {} & -0.618929 \times (\text{Murder}) \\ & - 0.170526 \times (\text{Rape}) \\ & + 0.047125 \times (\text{Robbery}) \\ & . \\ & . \\ & . \\ & + 0.504003 \times (\text{Auto\_Theft})\end{aligned}$$

where the variables are standardized.

Figure 62.4 Eigenvectors Table

Eigenvectors							
Variable	Prin1	Prin2	Prin3	Prin4	Prin5	Prin6	Prin7
Murder	0.30289	-0.61893	0.17353	-0.23308	0.54896	0.26371	0.26428
Rape	0.43410	-0.17053	-0.23539	0.06540	0.18075	-0.78232	-0.27946
Robbery	0.39705	0.04713	0.49208	-0.57470	-0.50808	-0.09452	-0.02497
Assault	0.39622	-0.35142	-0.05343	0.61743	-0.51525	0.17395	0.19921
Burglary	0.44164	0.20861	-0.22454	-0.02750	0.11273	0.52340	-0.65085
Larceny	0.35634	0.40570	-0.53681	-0.23231	0.02172	0.04085	0.60346
Auto_Theft	0.28834	0.50400	0.57524	0.41853	0.35939	-0.06024	0.15487

The first component is a measure of the overall crime rate, because the first eigenvector shows approximately equal loadings on all variables. The second eigenvector has high positive loadings on the variables Auto_Theft and Larceny and high negative loadings on the variables Murder and Assault. There is also a small positive loading on the variable Burglary and a small negative loading on the variable Rape. This component seems to measure the preponderance of property crime compared to violent crime. The interpretation of the third component is not obvious.

Syntax: HPPRINCOMP Procedure

The following statements are available in the HPPRINCOMP procedure:

PROC HPPRINCOMP < *options* > **;**
  BY *variables* **;**
  CODE < *options* > **;**
  FREQ *variable* **;**
  ID *variables* **;**
  OUTPUT < **OUT=***SAS-data-set* >
      < *keyword* < **=***prefix* > >. . . < *keyword* < **=***prefix* > > **;**
  PARTIAL *variables* **;**
  PERFORMANCE *performance-options* **;**
  VAR *variables* **;**
  WEIGHT *variable* **;**

The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC HPPRINCOMP statement. The remaining statements are described in alphabetical order.

PROC HPPRINCOMP Statement

PROC HPPRINCOMP < *options* > **;**

The PROC HPPRINCOMP statement invokes the HPPRINCOMP procedure. Optionally, it also identifies the input and output data sets, specifies the analyses to be performed, and controls displayed output. Table 62.1 summarizes the options available in the PROC HPPRINCOMP statement.

Table 62.1 PROC HPPRINCOMP Statement Options

Option	Description
Specify Data Sets	
DATA=	Specifies the name of the input data set
OUT=	Specifies the name of the output data set
OUTSTAT=	Specifies the name of the output data set that contains various statistics

Table 62.1 *continued*

Option	Description
Specify Details of Analysis	
COV	Computes the principal components from the covariance matrix
METHOD=	Specifies the principal component extraction method to be used
N=	Specifies the number of principal components to be computed
NOINT	Omits the intercept from the model
PREFIX=	Specifies a prefix for naming the principal components
PARPREFIX=	Specifies a prefix for naming the residual variables
SINGULAR=	Specifies the singularity criterion
STD	Standardizes the principal component scores
VARDEF=	Specifies the divisor used in calculating variances and standard deviations
Suppress the Display of Output	
NOPRINT	Suppresses the display of all output

The following list provides details about these *options*.

COVARIANCE

COV

computes the principal components from the covariance matrix. If you omit the COV option, the correlation matrix is analyzed. The COV option causes variables that have large variances to be more strongly associated with components that have large eigenvalues, and it causes variables that have small variances to be more strongly associated with components that have small eigenvalues. You should not specify the COV option unless the units in which the variables are measured are comparable or the variables are standardized in some way.

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed. The data set can only be an ordinary SAS data set (raw data). If you omit the DATA= option, the HPPRINCOMP procedure uses the most recently created SAS data set.

If PROC HPPRINCOMP 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 HPPRINCOMP 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*).

METHOD=EIG | ITERGS< (*iter-options***) > | NIPALS< (***iter-options***) >**

specifies the principal component extraction method to be used. You can specify the following values:

EIG

requests eigenvalue decomposition.

ITERGS< (*iter-options*) >

requests the iterative method based on Gram-Schmidt orthogonalization (ITERGS) of Andrecut (2009). You can also specify the following optional *iter-options* in parentheses after METHOD=ITERGS:

EPSILON=*n*

specifies the convergence criterion for the iterative method. By default, EPSILON=1E–12.

MAXITER=*n*

specifies the maximum number of iterations for the iterative method. By default, MAXITER=5000.

NOCENTER

suppresses centering of the numeric variables to be analyzed. This option is useful if the analysis variables are already centered and scaled.

NOSCALE

suppresses scaling of the numeric variables to be analyzed. This option is useful if the analysis variables are already centered and scaled.

NIPALS< (*iter-options*) >

requests the nonlinear iterative partial least squares (NIPALS) method. You can also specify the optional *iter-options* in parentheses after METHOD=NIPALS.

By default, METHOD=EIG. If you specify METHOD=NIPALS or METHOD=ITERGS, the following options in the PROC HPPRINCOMP statement are ignored: COV, NOINT, OUT=, OUTSTAT=, PARPREFIX=, SINGULAR=, and STD.

N=*number*

specifies the number of principal components to be computed. The default is the number of variables. The value of the N= option must be an integer greater than or equal to 0.

NOINT

omits the intercept from the model. In other words, the NOINT option requests that the covariance or correlation matrix not be corrected for the mean. When you specify the NOINT option in the HPPRINCOMP procedure, the covariance matrix and, hence, the standard deviations are not corrected for the mean. If you want to obtain the standard deviations corrected for the mean, you can obtain them by using a procedure such as PROC MEANS.

If you use the NOINT option and also create an OUTSTAT= data set, the data set is TYPE=UCORR or TYPE=UCOV rather than TYPE=CORR or TYPE=COV.

NOPRINT

suppresses the display of all output. This option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, “Using the Output Delivery System.”

OUT=*SAS-data-set*

creates an output SAS data set to contain observationwise principal component scores. To avoid data duplication when you have large data sets, the variables in the input data set are *not* included in the output data set; however, variables that are specified in the ID statement are included.

If the input data are in distributed form, in which access of data in a particular order cannot be guaranteed, the HPPRINCOMP procedure copies the distribution or partition key to the output data set so that its contents can be joined with the input data.

If you want to create a SAS data set in a permanent library, you must specify a two-level name. For more information about permanent libraries and SAS data sets, see *SAS Language Reference: Concepts*. For more information about OUT= data sets, see the section "Output Data Sets" on page 4887.

OUTSTAT=*SAS-data-set*

creates an output SAS data set to contain means, standard deviations, number of observations, correlations or covariances, eigenvalues, and eigenvectors. If you specify the COV option, the data set is TYPE=COV or TYPE=UCOV, depending on the NOINT option, and it contains covariances; otherwise, the data set is TYPE=CORR or TYPE=UCORR, depending on the NOINT option, and it contains correlations. If you specify the PARTIAL statement, the OUTSTAT= data set also contains R squares.

If you want to create a SAS data set in a permanent library, you must specify a two-level name. For more information about permanent libraries and SAS data sets, see *SAS Language Reference: Concepts*. For more information about OUTSTAT= data sets, see the section "Output Data Sets" on page 4887.

PREFIX=*name*

specifies a prefix for naming the principal components. By default, the *names* are Prin1, Prin2, ..., Prin*n*. If you specify PREFIX=Abc, the components are named Abc1, Abc2, Abc3, and so on. The number of characters in the prefix plus the number of digits required to designate the variables should not exceed the current name length that is defined by the VALIDVARNAME= system option.

PARPREFIX=*name*

PPREFIX=*name*

RPREFIX=*name*

specifies a prefix for naming the residual variables in the OUT= data set and the OUTSTAT= data set. By default, the prefix is R_. The number of characters in the prefix plus the maximum length of the variable names should not exceed the current name length that is defined by the VALIDVARNAME= system option.

SINGULAR=*p*

SING=*p*

specifies the singularity criterion, where $0 < p < 1$. If a variable in a PARTIAL statement has an R square as large as $1 - p$ when predicted from the variables listed before it in the statement, the variable is assigned a standardized coefficient of 0. By default, SINGULAR=1E–8.

STANDARD

STD

standardizes the principal component scores in the OUT= data set to unit variance. If you omit the STANDARD option, the scores have a variance equal to the corresponding eigenvalue. Note that the STANDARD option has no effect on the eigenvalues themselves.

VARDEF=DF | N | WDF | WEIGHT | WGT

specifies the divisor to be used in calculating variances and standard deviations. By default, VARDEF=DF. The following table displays the values and associated divisors:

Value	Divisor	Formula	
DF	Error degrees of freedom	$n - i$	(before partialing)
		$n - p - i$	(after partialing)
N	Number of observations	n	
WEIGHT | WGT	Sum of weights	$\sum_{j=1}^{n} w_j$	
WDF	Sum of weights minus one	$\left(\sum_{j=1}^{n} w_j\right) - i$	(before partialing)
		$\left(\sum_{j=1}^{n} w_j\right) - p - i$	(after partialing)

In the formulas for VARDEF=DF and VARDEF=WDF, p is the number of degrees of freedom of the variables in the PARTIAL statement, and i is 0 if the NOINT option is specified and 1 otherwise.

BY Statement

BY *variables* **;**

You can specify a BY statement in PROC HPPRINCOMP 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 in the HPPRINCOMP 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*.

CODE Statement

CODE < *options* > **;**

The CODE statement enables you to write SAS DATA step code for computing the principal component scores either to a file or to a catalog entry. This code can then be included in a DATA step to score new data.

The CODE statement is not supported when the PARTIAL statement is specified. If you specify more than one CODE statement, only the last one specified is used.

Table 62.3 summarizes the *options* available in the CODE statement.

Table 62.3 CODE Statement Options

Option	Description
CATALOG=	Names the catalog entry where the generated code is saved
FILE=	Names the file where the generated code is saved
FORMAT=	Specifies the numeric format for the eigenvectors
GROUP=	Specifies the group identifier for array names and statement labels
LINESIZE=	Specifies the line size of the generated code

For more information about the syntax of the CODE statement, see the section "CODE Statement" in Chapter 19, "Shared Concepts and Topics."

FREQ Statement

FREQ *variable* **;**

The *variable* in the FREQ statement identifies a numeric variable in the data set that contains the frequency of occurrence of each observation. SAS high-performance analytics procedures that support the FREQ statement treat each observation as if it appeared f times, where f is the value of the FREQ variable for the observation. If the frequency value is not an integer, it is truncated to an integer. If the frequency value is less than 1 or missing, the observation is not used in the analysis. When the FREQ statement is not specified, each observation is assigned a frequency of 1.

The FREQ statement is not supported if you specify METHOD=NIPALS or METHOD=ITERGS in the PROC HPPRINCOMP statement.

ID Statement

ID *variables* **;**

The ID statement lists one or more variables from the input data set that are transferred to output data sets created by SAS high-performance analytics 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 analytics procedures, see the section "ID Statement" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

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 PROC HPPRINCOMP fits the model. If you do not specify a *keyword*, then only the principal component scores are included.

The OUTPUT statement causes the OUT= option in the PROC HPPRINCOMP statement to be ignored.

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 that you specify in the ID statement are included. If the input data are in distributed form, in which accessing data in a particular order cannot be guaranteed, the HPPRINCOMP 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:

OUT=*SAS-data-set*

DATA=*SAS-data-set*

specifies the name of the output data set. If you omit this option, 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 HPPRINCOMP procedure assigns a default prefix based on the type of statistic requested. For example, for the VAR variables x1 and x2, RESIDUAL produces two residual value variables, R_x1 and R_x2.

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

H

requests the approximate leverage. The default prefix is H.

STD

requests standardized (centered and scaled) VAR variable values for each VAR variable. The default prefix is Std.

STDSSE

requests the sum of squares of residuals for standardized VAR variables. The default prefix is StdSSE.

TSQUARE

T2

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

RESIDUAL

RESID

R

requests residuals for each VAR variable. The default prefix is R.

SCORE

requests principal component scores for each principal component. The default prefix is Score.

If you specify METHOD=EIG, the only valid *keywords* are RESIDUAL (if you also specify the PARTIAL statement) and SCORE. Other *keywords* are ignored.

The output variables that contain the requested statistic are named as follows, according to the *keyword* that you specify:

- The *keywords* RESIDUAL and STD define an output variable for each VAR variable, so the variables that correspond to each VAR variable are named by appending the name of the VAR variable to the prefix. For example, if the model has the VAR variables x1 and x2, then RESIDUAL=R produces the variables R_x1 and R_x2.
- The *keyword* SCORE defines an output variable for each principal component, so the variables that correspond to each successive component are named by appending the component number to the prefix. For example, if the model has three principal components, then SCORE=T produces the variables T1, T2, and T3.
- The *keywords* H, STDSSE, and TSQUARE each define a single output variable, so the variable name matches the prefix.

PARTIAL Statement

PARTIAL *variables* **;**

If you want to analyze a partial correlation or covariance matrix, specify the names of the numeric variables to be partialed out in the PARTIAL statement. The HPPRINCOMP procedure computes the principal components of the residuals from the prediction of the VAR variables by the PARTIAL variables. If you request an OUT= or OUTSTAT= data set, the residual variables are named by prefixing either the characters R_ (by default) or the string specified in the PARPREFIX= option to the VAR variables.

The PARTIAL statement is not supported if you specify METHOD=NIPALS or METHOD=ITERGS in the PROC HPPRINCOMP statement.

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 HPPRINCOMP procedure.

You can also use the PERFORMANCE statement to control whether the HPPRINCOMP 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*).

VAR Statement

VAR *variables* **;**

The VAR statement lists the numeric variables to be analyzed. If you omit the VAR statement, all numeric variables that are not specified in other statements are analyzed.

WEIGHT Statement

WEIGHT *variable* **;**

The *variable* in the WEIGHT statement is used as a weight to perform a weighted analysis of the data. Observations that have nonpositive or missing weights are not included in the analysis. If you do not specify a WEIGHT statement, all observations that are used in the analysis are assigned a weight of 1.

The WEIGHT statement is not supported if you specify METHOD=NIPALS or METHOD=ITERGS in the PROC HPPRINCOMP statement.

Details: HPPRINCOMP Procedure

Computing Principal Components

The HPPRINCOMP procedure implements several algorithms to calculate principal components: eigenvalue decomposition, NIPALS, and ITERGS of Andrecut (2009). Eigenvalue decomposition is more efficient when you want to calculate all principal components, whereas the NIPALS method is faster if you want to extract only the first few principal components. For high-dimensional data sets, the NIPALS method is more efficient, whereas it gets expensive for eigenvalue decomposition to calculate all the components simultaneously.

Eigenvalue Decomposition

Let $\mathbf{X}$ be a centered and scaled data matrix that has k numerical variables. The eigenvalue decomposition method bases the component extraction on the eigenvalue decomposition of the covariance matrix $\mathbf{X}'\mathbf{X}$, which extracts all the k principal components simultaneously. Each principal component is a linear combination of the original variables, and each component is orthogonal, with coefficients equal to the eigenvectors of the covariance matrix $\mathbf{X}'\mathbf{X}$. The eigenvectors are usually normalized to have unit length. The principal components are sorted by descending order of the eigenvalues, which are equal to the variances of the components.

NIPALS

The nonlinear iterative partial least squares (NIPALS) method extracts the principal components successively based on the data matrix $\mathbf{X}$. The NIPALS method starts by calculating the loadings, $\mathbf{p}$, as $\mathbf{p}' = (\mathbf{t}'\mathbf{t})^{-1}\mathbf{t}'\mathbf{X}$, where $\mathbf{t}$ is the score vector. It then calculates an improved score vector, $\mathbf{t} = \mathbf{X}\mathbf{p}$. The method iteratively computes the improved $\mathbf{p}$ and $\mathbf{t}$ until convergence is reached.

This process accounts for how the first principal component is extracted. The second component is extracted in the same way, by replacing $\mathbf{X}$ with the residual from the first component: $\mathbf{E} = \mathbf{X} - \mathbf{t}\mathbf{p}'$.

For large data matrices or matrices that have a high degree of column collinearity, the NIPALS method suffers from loss of orthogonality because of the machine-precision errors that accumulate at each iteration step. In practice, the NIPALS method is used to extract only the first few principal components.

ITERGS

The iterative method based on Gram-Schmidt orthogonalization (ITERGS) of Andrecut (2009) overcomes the issue of loss of orthogonality in the NIPALS method by applying Gram-Schmidt reorthogonalization correction to both the loadings and the scores at each iteration step:

$$\begin{aligned}\mathbf{p}_c &= \mathbf{p} - \mathbf{P}_k\mathbf{P}'_k\mathbf{p} \\ \mathbf{t}_c &= \mathbf{t} - \mathbf{T}_k\mathbf{T}'_k\mathbf{t}\end{aligned}$$

Here, $\mathbf{p}_c$ and $\mathbf{t}_c$ are the corrected loading vector and score vector, respectively. $\mathbf{P}_k$ is the matrix that is formed by using the first k loadings. $\mathbf{T}_k$ is the matrix that is formed by using the first k scores.

The ITERGS method stabilizes the iterative process at the cost of increased computational effort.

Missing Values

Observations that have missing values for any variable in the VAR, PARTIAL, FREQ, or WEIGHT statement are omitted from the analysis and are given missing values for principal component scores in the OUT= data set.

Output Data Sets

When an observationwise output data set is created, many SAS procedures 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 can 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 you run PROC HPPRINCOMP in distributed mode, see the section "Output Data Sets" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

OUT= Data Set

The new variables that are created for the OUT= data set contain the principal component scores. The N= option determines the number of new variables. The names of the new variables are formed by concatenating the value given by the PREFIX= option (or Prin if PREFIX= is omitted) to the numbers 1, 2, 3, and so on. The new variables have mean 0 and a variance equal to the corresponding eigenvalue, unless you specify the STANDARD option to standardize the scores to unit variance. Also, if you specify the COV option, PROC

HPPRINCOMP computes the principal component scores from the corrected or uncorrected (if the NOINT option is specified) variables rather than from the standardized variables.

If you use a PARTIAL statement, the OUT= data set also contains the residuals from predicting the VAR variables from the PARTIAL variables.

OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR data set that the CORR procedure produces. The following table relates the TYPE= value for the OUTSTAT= data set to the options that are specified in the PROC HPPRINCOMP statement:

Options	**TYPE=**
(Default)	CORR
COV	COV
NOINT	UCORR
COV NOINT	UCOV

Note that the default (neither the COV nor NOINT option) produces a TYPE=CORR data set.

The new data set contains the following variables:

- the BY variables, if any
- two new variables, _TYPE_ and _NAME_, both character variables
- the variables that are analyzed (that is, those in the VAR statement); or, if there is no VAR statement, all numeric variables not listed in any other statement; or, if there is a PARTIAL statement, the residual variables as described in the section “OUT= Data Set”

Each observation in the new data set contains some type of statistic, as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are as follows:

TYPE **Contents**

MEAN mean of each variable. If you specify the PARTIAL statement, this observation is omitted.

STD standard deviations. If you specify the COV option, this observation is omitted, so the SCORE procedure does not standardize the variables before computing scores. If you use the PARTIAL statement, the standard deviation of a variable is computed as its root mean squared error as predicted from the PARTIAL variables.

USTD uncorrected standard deviations. When you specify the NOINT option in the PROC HPPRINCOMP statement, the OUTSTAT= data set contains standard deviations not corrected for the mean. However, if you also specify the COV option in the PROC HPPRINCOMP statement, this observation is omitted.

N number of observations on which the analysis is based. This value is the same for each variable. If you specify the PARTIAL statement and the value of the VARDEF= option is DF or unspecified, then the number of observations is decremented by the degrees of freedom for the PARTIAL variables.

SUMWGT

the sum of the weights of the observations. This value is the same for each variable. If you specify the PARTIAL statement and VARDEF=WDF, then the sum of the weights is decremented by the degrees of freedom for the PARTIAL variables. This observation is output only if the value is different from that in the observation for which _TYPE_='N'.

CORR

correlations between each variable and the variable specified by the _NAME_ variable. The number of observations for which _TYPE_='CORR' is equal to the number of variables being analyzed. If you specify the COV option, no _TYPE_='CORR' observations are produced. If you use the PARTIAL statement, the partial correlations, not the raw correlations, are output.

UCORR

uncorrected correlation matrix. When you specify the NOINT option without the COV option in the PROC HPPRINCOMP statement, the OUTSTAT= data set contains a matrix of correlations not corrected for the means. However, if you also specify the COV option in the PROC HPPRINCOMP statement, this observation is omitted.

COV

covariances between each variable and the variable specified by the _NAME_ variable. _TYPE_='COV' observations are produced only if you specify the COV option. If you use the PARTIAL statement, the partial covariances, not the raw covariances, are output.

UCOV

uncorrected covariance matrix. When you specify the NOINT and COV options in the PROC HPPRINCOMP statement, the OUTSTAT= data set contains a matrix of covariances not corrected for the means.

EIGENVAL

eigenvalues. If the N= option requests less than the maximum number of principal components, only the specified number of eigenvalues are produced, with missing values filling out the observation.

SCORE

eigenvectors. The _NAME_ variable contains the name of the corresponding principal component as constructed from the PREFIX= option. The number of observations for which _TYPE_='SCORE' equals the number of principal components computed. The eigenvectors have unit length unless you specify the STD option, in which case the unit-length eigenvectors are divided by the square roots of the eigenvalues to produce scores that have unit standard deviations.

To obtain the principal component scores, if the COV option is not specified, these coefficients should be multiplied by the standardized data. For the COV option, these coefficients should be multiplied by the centered data. To center and standardize the data, you should use means that are obtained from the observation for which _TYPE_='MEAN' and standard deviations that are obtained from the observation for which _TYPE_='STD'.

USCORE

scoring coefficients to be applied without subtracting the mean from the raw variables. Observations for which _TYPE_='USCORE' are produced when you specify the NOINT option in the PROC HPPRINCOMP statement.

To obtain the principal component scores, these coefficients should be multiplied by the data that are standardized by the uncorrected standard deviations obtained from the observation for which _TYPE_='USTD'.

RSQUARED

R squares for each VAR variable as predicted by the PARTIAL variables.

B

regression coefficients for each VAR variable as predicted by the PARTIAL variables. This observation is produced only if you specify the COV option.

STB

standardized regression coefficients for each VAR variable as predicted by the PARTIAL variables. If you specify the COV option, this observation is omitted.

You can use the data set in the SCORE procedure to compute principal component scores, or you can use it as input to the FACTOR procedure and specify METHOD=SCORE to rotate the components. If you use the PARTIAL statement, the scoring coefficients should be applied to the residuals, not to the original variables.

Computational Method

Multithreading

Threading is 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 is the concurrent execution of threads. When multithreading is possible, you can realize substantial performance gains compared to the performance that you get from sequential (single-threaded) execution.

The number of threads that the HPPRINCOMP 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 HPPRINCOMP procedure schedules threads as if it were executing on a system that had 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 HPPRINCOMP procedure allocates one thread per CPU.

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

- formation of the crossproducts matrix
- computation of loadings, scores, and residual sums of squares
- principal component scoring of observations

Displayed Output

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

Model Information

The “Model Information” table displays basic information about the model, including the input data set and the principal component extraction method that is used in the analysis.

Number of Observations

The “Number of Observations” table displays the number of observations read from the input data set and the number of observations used in the analysis. If you specify a FREQ statement, the table also displays the sum of frequencies read and used.

Number of Variables

The “Number of Variables” table displays the number of VAR variables, the number of PARTIAL variables, and the number of principal components to be extracted.

Simple Statistics

If you specify METHOD=EIG, the HPPRINCOMP procedure produces a “Simple Statistics” table that displays the mean and standard deviation (std) for each variable. If you specify the NOINT option, the uncorrected standard deviation (ustd) is displayed.

Centering and Scaling Information

If you specify METHOD=NIPALS or METHOD=ITERGS, the HPPRINCOMP procedure produces a “Centering and Scaling Information” table that displays the centering and scaling information for each variable.

Explained Variation of Variables

If you specify METHOD=NIPALS or METHOD=ITERGS, the HPPRINCOMP procedure produces an “Explained Variation of Variables” table that displays the fraction of variation that is accounted for in each variable by each successive principal component.

Correlation Matrix

If you specify METHOD=EIG, the HPPRINCOMP procedure produces a "Correlation Matrix" table that displays the correlation or, if you specify the COV option, the covariance matrix.

Regression Statistics

When you specify the PARTIAL statement, the HPPRINCOMP procedure produces a "Regression Statistics" table that displays the R square and root mean squared error (RMSE) for each VAR variable as predicted by the PARTIAL variables.

Regression Coefficients

When you specify the PARTIAL statement, the HPPRINCOMP procedure produces a "Regression Coefficients" table that displays standardized regression coefficients or, if you specify the COV option, regression coefficients for predicting the VAR variables from the PARTIAL variables.

Partial Correlation Matrix

When you specify the PARTIAL statement, the HPPRINCOMP procedure produces a "Partial Correlation Matrix" table that displays the partial correlation matrix or, if you specify the COV option, the partial covariance matrix.

Total Variance

If you specify METHOD=EIG and the COV option, the HPPRINCOMP procedure produces a simple table that displays the total variance.

Eigenvalues

The "Eigenvalues" table displays eigenvalues of the correlation or covariance matrix (if you specify METHOD=EIG) or eigenvalues of the data matrix (if you specify METHOD=NIPALS or METHOD=ITERGS), along with the difference between successive eigenvalues, the proportion of variance explained by each eigenvalue, and the cumulative proportion of variance explained.

Eigenvectors

If you specify METHOD=EIG, the HPPRINCOMP procedure produces an "Eigenvectors" table that displays the eigenvectors.

Loadings

If you specify METHOD=NIPALS or METHOD=ITERGS, the HPPRINCOMP procedure produces a "Loadings" table that displays the loadings.

Timing Information

If you specify the DETAILS option in the PERFORMANCE statement, the HPPRINCOMP procedure produces a "Timing" table that displays the elapsed time of each main task of the procedure.

ODS Table Names

PROC HPPRINCOMP assigns a name to each table that it creates. You can use these names to reference the ODS table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 62.4. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 62.4 ODS Tables Produced by PROC HPPRINCOMP

Table Name	Description	Required Statement / Option
CenScaleInfo	Centering and scaling information	METHOD=NIPALS | ITERGS
Corr	Correlation matrix	METHOD=EIG
Cov	Covariance matrix	METHOD=EIG and COV
DataAccessInfo	Information about modes of data access	Default output
Eigenvalues	Eigenvalues	Default output
Eigenvectors	Eigenvectors	METHOD=EIG
Loadings	Loadings	METHOD=NIPALS | ITERGS
ModelInfo	Model information	Default output
NObs	Number of observations read and used	Default output
NVars	Number of variables, partial variables, and principal components	Default output
ParCorr	Partial correlation matrix	PARTIAL statement
ParCov	Uncorrected partial covariance matrix	PARTIAL statement and COV
PerformanceInfo	Information about the high-performance computing environment	Default output
RegCoef	Regression coefficients	PARTIAL statement and COV
RSquareRMSE	Regression statistics: R-squares and RMSEs	PARTIAL statement
SimpleStatistics	Simple statistics	METHOD=EIG
StdRegCoef	Standardized regression coefficients	PARTIAL statement
Timing	Absolute and relative times of tasks that are performed by the procedure	DETAILS option in PERFORMANCE statement
TotalVariance	Total variance	METHOD=EIG and COV
Variation	Explained variation of variables	METHOD=NIPALS | ITERGS

Examples: HPPRINCOMP Procedure

Example 62.1: Analyzing Mean Temperatures of US Cities

This example analyzes mean daily temperatures of selected US cities in January and July. The following statements create the Temperature data set:

```
data Temperature;
   length Cityid $ 2;
   title 'Mean Temperature in January and July for Selected Cities ';
   input City $1-15 January July;
   Cityid = substr(City,1,2);
   datalines;
Mobile          51.2 81.6
Phoenix         51.2 91.2
Little Rock     39.5 81.4
Sacramento      45.1 75.2
Denver          29.9 73.0

   ... more lines ...

Cheyenne        26.6 69.1
;
```

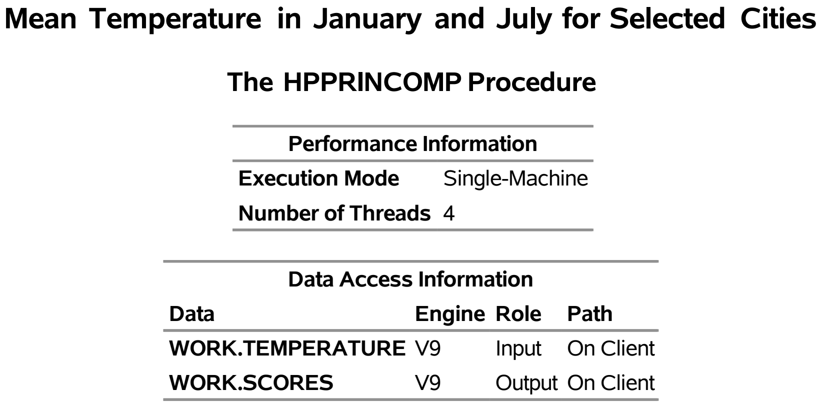
The following statements invoke the HPPRINCOMP procedure, which requests a principal component analysis of the Temperature data set and outputs the scores to the Scores data set (OUT= Scores). The Cityid variable in the ID statement is also included in the output data set.

```
title 'Mean Temperature in January and July for Selected Cities';
proc hpprincomp data=Temperature cov out=Scores;
   var July January;
   id Cityid;
run;
```

Output 62.1.1 displays the PROC HPPRINCOMP output. The standard deviation of January (11.712) is higher than the standard deviation of July (5.128). The COV option in the PROC HPPRINCOMP statement requests that the principal components be computed from the covariance matrix. The total variance is 163.474. The first principal component accounts for about 94% of the total variance, and the second principal component accounts for only about 6%. The eigenvalues sum to the total variance.

Note that January receives a higher loading on Prin1 because it has a higher standard deviation than July. Also note that the HPPRINCOMP procedure calculates the scores by using the centered variables rather than the standardized variables.

Output 62.1.1 Results of Principal Component Analysis



Mean Temperature in January and July for Selected Cities

The HPPRINCOMP Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

Data	Engine	Role	Path
WORK.TEMPERATURE	V9	Input	On Client
WORK.SCORES	V9	Output	On Client

Output 62.1.1 *continued*

Model Information	
Data Source	WORK.TEMPERATURE
Component Extraction Method	Eigenvalue Decomposition

Number of Observations Read	64
Number of Observations Used	64

Number of Variables	2
Number of Principal Components	2

Simple Statistics

Variable	Mean	Standard Deviation
July	75.60781	5.12762
January	32.09531	11.71243

Covariance Matrix

Variable	July	January
July	26.29248	46.82829
January	46.82829	137.18109

Total Variance	163.47356647

Eigenvalues of the Covariance Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	154.310607	145.147647	0.9439	0.9439
2	9.162960		0.0561	1.0000

Eigenvectors

Variable	Prin1	Prin2
July	0.34353	0.93914
January	0.93914	-0.34353

Example 62.2: Computing Principal Components in Single-Machine and Distributed Modes

PROC HPPRINCOMP shows its real power when the computation is conducted with multiple threads or in a distributed environment. This example shows how you can run PROC HPPRINCOMP in single-machine and distributed modes. For more information about the execution modes of SAS high-performance analytics 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 HPPRINCOMP. The following DATA step generates the data:

```
data ex2Data;
   array x{100};
   do i = 1 to 5000000;
      do j = 1 to dim(x);
         x[j] = ranuni(1);
      end;
      output;
   end;
run;
```

The following statements use PROC HPPRINCOMP to perform a principal component analysis and to output various statistics to the Stats data set (OUTSTAT= Stats):

```
   proc hpprincomp data=ex2Data n=20 outstat=Stats;
      var x:;
      performance details;
   run;
```

Output 62.2.1 shows the "Performance Information" table. This table shows that the HPPRINCOMP procedure executes in single-machine mode on four threads, because 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 62.2.1 Performance Information in Single-Machine Mode

The HPPRINCOMP Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	4

Output 62.2.2 shows timing information for the PROC HPPRINCOMP 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 reading the data and computing the moments.

Output 62.2.2 Timing in Single-Machine Mode

Procedure Task Timing

Task	Seconds	Percent
Reading Data and Computing Moments	53.02	85.98%
Computing Principal Components	8.63	13.99%
Producing Output Statistics Data Set	0.01	0.02%

To switch to running PROC HPPRINCOMP 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 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*).

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 hpprincomp data=ex2Data n=20 outstat=Stats;
   var x:;
   performance details nodes = 4
               host="&GRIDHOST" install="&GRIDINSTALLLOC";
run;
```

The execution mode in the "Performance Information" table shown in Output 62.2.3 indicates that the calculations were performed in a distributed environment that uses four nodes, each of which uses 32 threads.

Output 62.2.3 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 in the SAS log, which is issued by all high-performance analytics procedures:

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

Output 62.2.4 shows timing information for this distributed run of the HPPRINCOMP procedure. In contrast with the single-machine mode (where reading the data and computing the moments dominate the time spent), the majority of time in the distributed-mode run is spent distributing the data.

Output 62.2.4 Timing in Distributed Mode

Procedure Task Timing

Task	Seconds	Percent
Obtaining Settings	0.00	0.00%
Distributing Data	35.26	82.86%
Reading Data and Computing Moments	5.58	13.10%
Computing Principal Components	1.37	3.22%
Producing Output Statistics Data Set	0.05	0.11%
Waiting on Client	0.30	0.71%

Example 62.3: Extracting Principal Components with NIPALS

This example demonstrates the NIPALS method in PROC HPPRINCOMP, which extracts principal components successively. The data that this example uses are from the Getting Started section; they provide crime rates per 100,000 people in seven categories for each of the 50 US states in 1977. The following DATA step generates the data:

```
data Crime;
   title 'Crime Rates per 100,000 Population by State';
   input State $1-15 Murder Rape Robbery Assault
         Burglary Larceny Auto_Theft;
   datalines;
Alabama        14.2 25.2  96.8 278.3 1135.5 1881.9 280.7
Alaska         10.8 51.6  96.8 284.0 1331.7 3369.8 753.3
Arizona         9.5 34.2 138.2 312.3 2346.1 4467.4 439.5
Arkansas        8.8 27.6  83.2 203.4  972.6 1862.1 183.4
California     11.5 49.4 287.0 358.0 2139.4 3499.8 663.5

   ... more lines ...

Wisconsin       2.8 12.9  52.2  63.7  846.9 2614.2 220.7
Wyoming          .  21.9  39.7 173.9  811.6 2772.2 282.0
;
```

The following statements use PROC HPPRINCOMP to extract principal components by using the NIPALS method:

```
proc hpprincomp data=Crime method=nipals;
run;
```

Output 62.3.1 displays the PROC HPPRINCOMP output. The "Model Information" table shows that the NIPALS method is used to extract principal components. The "Explained Variation of Variables" table lists the fraction of variation that is accounted for in each variable by each of the seven principal components. All the variation in each variable is accounted for by seven principal components because there are only seven variables. The eigenvalues indicate that two or three components provide a good summary of the data: two components account for 76% of the total variance, and three components account for 87%. Subsequent components account for less than 5% each.

Note that in the Getting Started section, the principal components are extracted from the same data by using the eigenvalue decomposition method; the "Eigenvalues" table generated there matches the one generated by the NIPALS method. Also, the eigenvectors in the "Eigenvectors" table match the loading factors in the "Loadings" table.

Output 62.3.1 Results of Principal Component Analysis Using NIPALS

Crime Rates per 100,000 Population by State

The HPPRINCOMP Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	4

Data Access Information

Data	Engine	Role	Path
WORK.CRIME	V9	Input	On Client

Model Information	
Data Source	WORK.CRIME
Component Extraction Method	NIPALS

Number of Observations Read	50
Number of Observations Used	48

Output 62.3.1 *continued*

Number of Variables	7
Number of Principal Components	7

Centering and Scaling Information

Variable	Subtracted off	Divided by
Murder	7.51667	3.93059
Rape	26.07500	10.81304
Robbery	127.55625	88.49374
Assault	214.58750	100.64360
Burglary	1316.37917	423.31261
Larceny	2696.88542	714.75023
Auto_Theft	383.97917	194.37033

Explained Variation of Variables

Variable	Prin1	Prin2	Prin3	Prin4	Prin5	Prin6	Prin7
Murder	0.37117	0.85539	0.87790	0.89562	0.97555	0.99143	1.00000
Rape	0.76242	0.79917	0.84059	0.84199	0.85065	0.99041	1.00000
Robbery	0.63783	0.64064	0.82164	0.92942	0.99788	0.99992	1.00000
Assault	0.63517	0.79127	0.79341	0.91781	0.98822	0.99513	1.00000
Burglary	0.78913	0.84414	0.88183	0.88207	0.88544	0.94800	1.00000
Larceny	0.51373	0.72178	0.93718	0.95479	0.95492	0.95530	1.00000
Auto_Theft	0.33638	0.65746	0.90481	0.96197	0.99623	0.99706	1.00000

Eigenvalues of the Data Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	4.045824	2.781795	0.5780	0.5780
2	1.264030	0.516529	0.1806	0.7586
3	0.747500	0.421175	0.1068	0.8653
4	0.326325	0.061119	0.0466	0.9120
5	0.265207	0.036843	0.0379	0.9498
6	0.228364	0.105613	0.0326	0.9825
7	0.122750		0.0175	1.0000

Loadings

Variable	Prin1	Prin2	Prin3	Prin4	Prin5	Prin6	Prin7
Murder	0.30289	-0.61893	0.17353	-0.23308	0.54896	-0.26371	-0.26428
Rape	0.43410	-0.17053	-0.23539	0.06540	0.18075	0.78232	0.27946
Robbery	0.39705	0.04713	0.49208	-0.57470	-0.50808	0.09452	0.02497
Assault	0.39622	-0.35142	-0.05343	0.61744	-0.51525	-0.17395	-0.19921
Burglary	0.44164	0.20861	-0.22454	-0.02750	0.11273	-0.52340	0.65085
Larceny	0.35634	0.40570	-0.53681	-0.23231	0.02172	-0.04085	-0.60346
Auto_Theft	0.28834	0.50400	0.57524	0.41853	0.35939	0.06024	-0.15487

References

Andrecut, M. (2009). “Parallel GPU Implementation of Iterative PCA Algorithms.” *Journal of Computational Biology* 16:1593–1599.

Cooley, W. W., and Lohnes, P. R. (1971). *Multivariate Data Analysis*. New York: John Wiley & Sons.

Gnanadesikan, R. (1977). *Methods for Statistical Data Analysis of Multivariate Observations*. New York: John Wiley & Sons.

Hotelling, H. (1933). “Analysis of a Complex of Statistical Variables into Principal Components.” *Journal of Educational Psychology* 24:417–441, 498–520.

Kshirsagar, A. M. (1972). *Multivariate Analysis*. New York: Marcel Dekker.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979). *Multivariate Analysis*. London: Academic Press.

Morrison, D. F. (1976). *Multivariate Statistical Methods*. 2nd ed. New York: McGraw-Hill.

Pearson, K. (1901). “On Lines and Planes of Closest Fit to Systems of Points in Space.” *Philosophical Magazine* 6:559–572.

Rao, C. R. (1964). “The Use and Interpretation of Principal Component Analysis in Applied Research.” *Sankhyā, Series A* 26:329–358.

Wold, S., Esbensen, K., and Geladi, P. (1987). “Principal Component Analysis.” *Chemometrics and Intelligent Laboratory Systems* 2:37–52.

Subject Index

Syntax Index