THE POWER TO KNOW.

SAS/STAT® 15.1 User's Guide
The HPCANDISC 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 54
The HPCANDISC Procedure

Contents

Overview: HPCANDISC Procedure

The HPCANDISC procedure is a high-performance procedure that performs canonical discriminant analysis. It is a high-performance version of the CANDISC procedure in SAS/STAT software. PROC HPCANDISC runs in either single-machine mode or distributed mode.

NOTE: Distributed mode requires SAS High-Performance Statistics.

Canonical discriminant analysis is a dimension-reduction technique related to principal component analysis and canonical correlation. The methodology that is used in deriving the canonical coefficients parallels that of a one-way multivariate analysis of variance (MANOVA). MANOVA tests for equality of the mean vector across class levels. Canonical discriminant analysis finds linear combinations of the quantitative variables that provide maximal separation between classes or groups. Given a classification variable and several quantitative variables, the HPCANDISC procedure derives *canonical variables*, which are linear combinations of the quantitative variables that summarize between-class variation in much the same way that principal components summarize total variation.

The HPCANDISC procedure performs a canonical discriminant analysis, computes squared Mahalanobis distances between class means, and performs both univariate and one-way multivariate analyses of variance. Two output data sets can be produced: one that contains the canonical coefficients and another that contains, among other things, scored canonical variables. You can rotate the canonical coefficients output data set by using the FACTOR procedure. It is customary to standardize the canonical coefficients so that the canonical variables have means that are equal to 0 and pooled within-class variances that are equal to 1. PROC HPCANDISC displays both standardized and unstandardized canonical coefficients. Correlations between the canonical variables and the original variables in addition to the class means for the canonical variables are also displayed; these correlations, sometimes known as loadings, are called *canonical structures*.

When you have two or more groups of observations that have measurements on several quantitative variables, canonical discriminant analysis derives a linear combination of the variables that has the highest possible multiple correlation with the groups. This maximal multiple correlation is called the *first canonical correlation*. The coefficients of the linear combination are the *canonical coefficients* or *canonical weights*. The variable that is defined by the linear combination is the *first canonical variable* or *canonical component*. The second canonical correlation is obtained by finding the linear combination uncorrelated with the first canonical variable that has the highest possible multiple correlation with the groups. The process of extracting canonical variables can be repeated until the number of canonical variables equals the number of original variables or the number of classes minus one, whichever is smaller.

The first canonical correlation is at least as large as the multiple correlation between the groups and any of the original variables. If the original variables have high within-group correlations, the first canonical correlation can be large even if all the multiple correlations are small. In other words, the first canonical variable can show substantial differences between the classes, even if none of the original variables do. Canonical variables are sometimes called *discriminant functions*, but this usage is ambiguous because the DISCRIM procedure produces very different functions for classification that are also called discriminant functions.

For each canonical correlation, PROC HPCANDISC tests the hypothesis that it and all smaller canonical correlations are zero in the population. An F approximation (Rao 1973; Kshirsagar 1972) is used that gives better small-sample results than the usual chi-square approximation. The variables should have an approximate multivariate normal distribution within each class, with a common covariance matrix in order for the probability levels to be valid.

Canonical discriminant analysis is equivalent to canonical correlation analysis between the quantitative variables and a set of dummy variables coded from the CLASS variable. Performing canonical discriminant analysis is also equivalent to performing the following steps:

1. Transform the variables so that the pooled within-class covariance matrix is an identity matrix.

2. Compute class means on the transformed variables.
3. Perform a principal component analysis on the means, weighting each mean by the number of observations in the class. The eigenvalues are equal to the ratio of between-class variation to within-class variation in the direction of each principal component.
4. Back-transform the principal components into the space of the original variables to obtain the canonical variables.

An interesting property of the canonical variables is that they are uncorrelated whether the correlation is calculated from the total sample or from the pooled within-class correlations. However, the canonical coefficients are not orthogonal, so the canonical variables do not represent perpendicular directions through the space of the original variables.

PROC HPCANDISC Features

The main features of the HPCANDISC procedure are as follows:

- performs a canonical discriminant analysis, computes squared Mahalanobis distances between class means, and performs both univariate and multivariate one-way analyses of variance
- can perform analysis on a massively parallel SAS high-performance appliance
- reads input data in parallel and writes output data in parallel when the data source is the appliance database
- is highly multithreaded during calculations of the within-class sum-of-squares-and-crossproducts (SSCP) matrix and the canonical variable scores
- supports a FREQ statement for grouped analysis
- supports a WEIGHT statement for weighted analysis
- displays both standardized and unstandardized canonical coefficients
- displays correlations between the canonical variables and the original variables
- displays class means for the canonical variables
- produces two output data sets: one that contains the canonical coefficients and another that contains scored canonical variables

PROC HPCANDISC Compared with PROC CANDISC

The HPCANDISC procedure and the CANDISC procedure in SAS/STAT have the following similarities and differences:

- All the statements that are available in PROC CANDISC are supported in the HPCANDISC procedure.
- As input, PROC CANDISC can accept ordinary SAS data set and other types of special SAS data sets. In the HPCANDISC procedure, only the ordinary SAS data set (raw data) can be used as input.
- The HPCANDISC procedure supports an ID statement that is not available in PROC CANDISC.
- The HPCANDISC procedure is specifically designed to operate in the high-performance distributed environment. By default, PROC HPCANDISC performs computations on multiple threads. The CANDISC procedure executes on a single thread.

Getting Started: HPCANDISC Procedure

The data in this example are measurements of 159 fish caught in Finland's Lake Laengelmaevesi; this data set is available from the Puranen. For each of the seven species (bream, roach, whitefish, parkki, perch, pike, and smelt), the weight, length, height, and width of each fish are tallied. Three different length measurements are recorded: from the nose of the fish to the beginning of its tail, from the nose to the notch of its tail, and from the nose to the end of its tail. The height and width are recorded as percentages of the third length variable. The fish data set is available from the Sashelp library.

The following step uses PROC HPCANDISC to find the three canonical variables that best separate the species of fish in the Sashelp.Fish data and create the output data set outcan. When the NCAN=3 option is specified, only the first three canonical variables are displayed. The ID statement adds the variable Species from the input data set to the output data set. The ODS EXCLUDE statement excludes the canonical structure tables and most of the canonical coefficient tables in order to obtain a more compact set of results. The TEMPLATE and SGRENDER procedures create a plot of the first two canonical variables. The following statements produce Figure 54.1 through Figure 54.6:

```
title 'Fish Measurement Data';

proc hpcandisc data=sashelp.fish ncan=3 out=outcan;
   ods exclude tstruc bstruc pstruc tcoef pcoef;
   id Species;
   class Species;
   var Weight Length1 Length2 Length3 Height Width;
run;

proc template;
   define statgraph scatter;
      begingraph;
         entrytitle 'Fish Measurement Data';
         layout overlayequated / equatetype=fit
            xaxisopts=(label='Canonical Variable 1')
            yaxisopts=(label='Canonical Variable 2');
            scatterplot x=Can1 y=Can2 / group=species name='fish';
            layout gridded / autoalign=(topright);
               discretelegend 'fish' / border=false opaque=false;
            endlayout;
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=outcan template=scatter;
run;
```

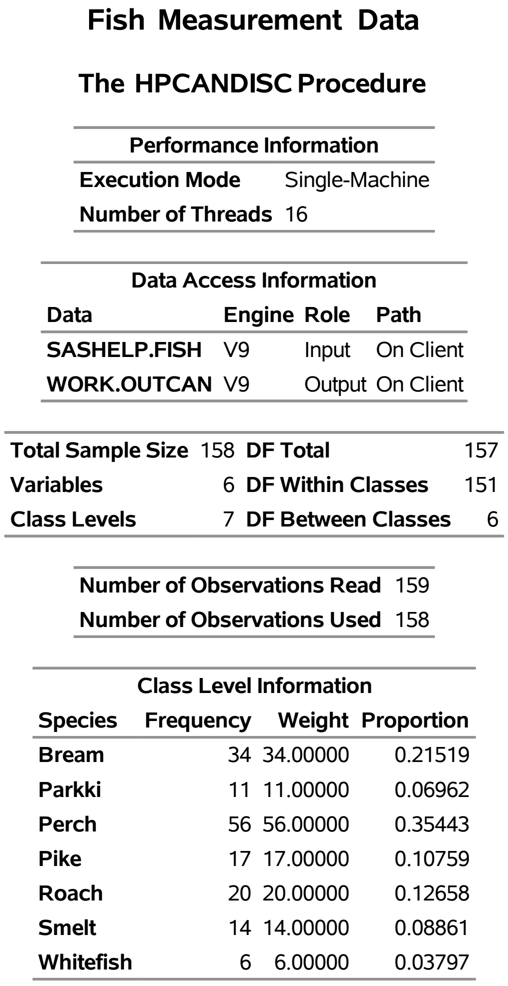
PROC HPCANDISC begins by displaying performance information, data access information, and summary information about the variables in the analysis, as shown in Figure 54.1.

The “Performance Information” table shows the procedure executes in single-machine mode; that is, the data reside and the computation is conducted on the machine where the SAS session executes. This run of the HPCANDISC procedure took place on a multicore machine that had four CPUs; one computational thread was spawned per CPU.

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

The summary information includes the number of observations, the number of quantitative variables in the analysis (specified using the VAR statement), and the number of class levels in the classification variable (specified using the CLASS statement). The value and frequency of each class level are also displayed.

Figure 54.1 Fish Data: Performance, Data Access, and Summary Information



Fish Measurement Data

The HPCANDISC Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	16

Data Access Information

Data	Engine	Role	Path
SASHELP.FISH	V9	Input	On Client
WORK.OUTCAN	V9	Output	On Client

Total Sample Size	158	**DF Total**	157
Variables	6	**DF Within Classes**	151
Class Levels	7	**DF Between Classes**	6

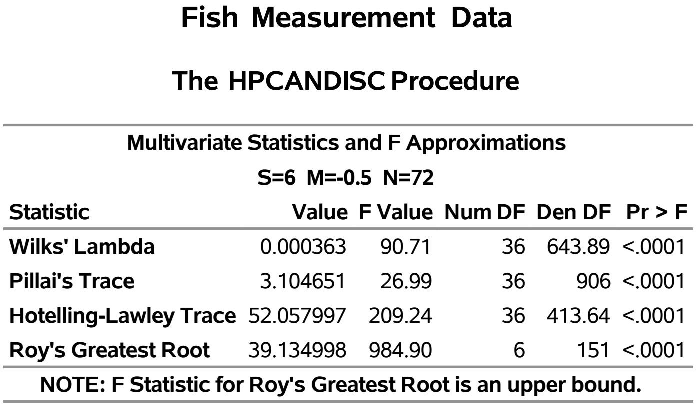
Number of Observations Read	159
Number of Observations Used	158

Class Level Information

Species	Frequency	Weight	Proportion
Bream	34	34.00000	0.21519
Parkki	11	11.00000	0.06962
Perch	56	56.00000	0.35443
Pike	17	17.00000	0.10759
Roach	20	20.00000	0.12658
Smelt	14	14.00000	0.08861
Whitefish	6	6.00000	0.03797

Figure 54.2 displays the “Multivariate Statistics and F Approximations” table. PROC HPCANDISC performs a one-way multivariate analysis of variance (one-way MANOVA) and provides four multivariate tests of the hypothesis that the class mean vectors are equal. These tests indicate that not all the mean vectors are equal ($p < 0.0001$).

Figure 54.2 Fish Data: MANOVA and Multivariate Tests



Fish Measurement Data

The HPCANDISC Procedure

Multivariate Statistics and F Approximations

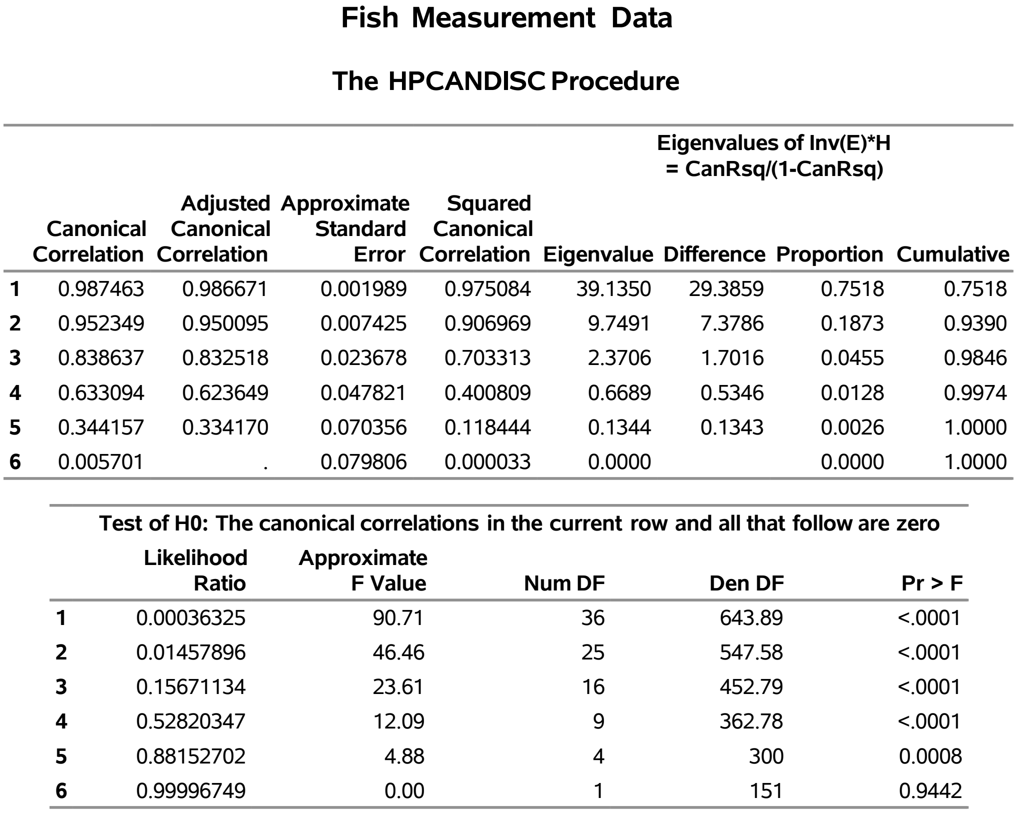
S=6 M=-0.5 N=72

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.000363	90.71	36	643.89	<.0001
Pillai's Trace	3.104651	26.99	36	906	<.0001
Hotelling-Lawley Trace	52.057997	209.24	36	413.64	<.0001
Roy's Greatest Root	39.134998	984.90	6	151	<.0001

NOTE: F Statistic for Roy's Greatest Root is an upper bound.

Figure 54.3 displays the “Canonical Correlations” table. The first canonical correlation is the greatest possible multiple correlation with the classes that you can achieve by using a linear combination of the quantitative variables. The first canonical correlation, displayed in the table, is 0.987463. The figure shows a likelihood ratio test of the hypothesis that the current canonical correlation and all smaller ones are zero. The first line is equivalent to Wilks’ lambda multivariate test.

Figure 54.3 Fish Data: Canonical Correlations



Fish Measurement Data

The HPCANDISC Procedure

					Eigenvalues of Inv(E)*H = CanRsq/(1-CanRsq)			
	Canonical Correlation	Adjusted Canonical Correlation	Approximate Standard Error	Squared Canonical Correlation	Eigenvalue	Difference	Proportion	Cumulative
1	0.987463	0.986671	0.001989	0.975084	39.1350	29.3859	0.7518	0.7518
2	0.952349	0.950095	0.007425	0.906969	9.7491	7.3786	0.1873	0.9390
3	0.838637	0.832518	0.023678	0.703313	2.3706	1.7016	0.0455	0.9846
4	0.633094	0.623649	0.047821	0.400809	0.6689	0.5346	0.0128	0.9974
5	0.344157	0.334170	0.070356	0.118444	0.1344	0.1343	0.0026	1.0000
6	0.005701	.	0.079806	0.000033	0.0000		0.0000	1.0000

Test of H0: The canonical correlations in the current row and all that follow are zero

	Likelihood Ratio	Approximate F Value	Num DF	Den DF	Pr > F
1	0.00036325	90.71	36	643.89	<.0001
2	0.01457896	46.46	25	547.58	<.0001
3	0.15671134	23.61	16	452.79	<.0001
4	0.52820347	12.09	9	362.78	<.0001
5	0.88152702	4.88	4	300	0.0008
6	0.99996749	0.00	1	151	0.9442

Figure 54.4 displays the “Raw Canonical Coefficients” table. The first canonical variable, Can1, shows that the linear combination of the centered variables Can1 = –0.0006 × Weight – 0.33 × Length1 2.49 × Length2 + 2.60 × Length3 + 1.12 × Height – 1.45 × Width separates the species most effectively.

Figure 54.4 Fish Data: Raw Canonical Coefficients

Fish Measurement Data

The HPCANDISC Procedure

Raw Canonical Coefficients

Variable	Can1	Can2	Can3
Weight	-0.00064851	-0.00523	-0.00560
Length1	-0.32944	-0.62660	-2.93432
Length2	-2.48613	-0.69025	4.04504
Length3	2.59565	1.80318	-1.13926
Height	1.12198	-0.71475	0.28320
Width	-1.44639	-0.90703	0.74149

Figure 54.5 displays the “Class Means on Canonical Variables” table. PROC HPCANDISC computes the means of the canonical variables for each class. The first canonical variable is the linear combination of the variables Weight, Length1, Length2, Length3, Height, and Width that provides the greatest difference (in terms of a univariate *F* test) between the class means. The second canonical variable provides the greatest difference between class means while being uncorrelated with the first canonical variable.

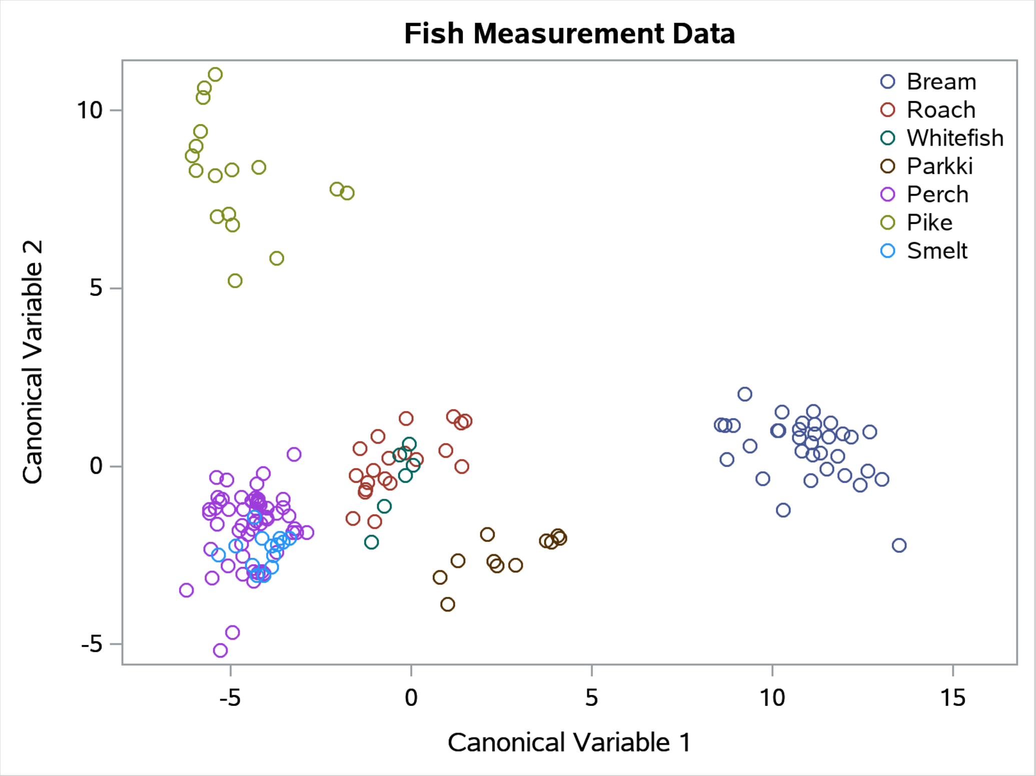
Figure 54.5 Fish Data: Class Means for Canonical Variables

Class Means on Canonical Variables

Species	Can1	Can2	Can3
Bream	10.94142	0.52078	0.23497
Parkki	2.58904	-2.54722	-0.49326
Perch	-4.47181	-1.70823	1.29281
Pike	-4.89689	8.22141	-0.16469
Roach	-0.35837	0.08734	-1.10056
Smelt	-4.09137	-2.35806	-4.03836
Whitefish	-0.39542	-0.42072	1.06459

Figure 54.6 displays a plot of the first two canonical variables, which shows that Can1 discriminates among three groups: (1) bream; (2) whitefish, roach, and parkki; and (3) smelt, pike, and perch. Can2 best discriminates between pike and the other species.

Figure 54.6 Fish Data: Plot of First Two Canonical Variables



Syntax: HPCANDISC Procedure

The following statements are available in the HPCANDISC procedure:

PROC HPCANDISC < *options* > ;
  BY *variables* ;
  CLASS *variable* ;
  FREQ *variable* ;
  ID *variables* ;
  PERFORMANCE *performance-options* ;
  VAR *variables* ;
  WEIGHT *variable* ;

The PROC HPCANDISC statement and a single CLASS statement are required. All other statements are optional.

PROC HPCANDISC Statement

PROC HPCANDISC < *options* > **;**

The PROC HPCANDISC statement invokes the HPCANDISC procedure. Optionally, it also identifies input and output data sets, specifies the analyses performed, and controls displayed output. Table 54.1 summarizes the options available in the PROC HPCANDISC statement.

Table 54.1 PROC HPCANDISC Statement Options

Option	Description
Specify Data Sets	
DATA=	Specifies the input data set
OUT=	Specifies the output data set that contains canonical scores
OUTSTAT=	Specifies the output statistics data set
Specify Details of Analysis	
NCAN=	Specifies the number of canonical variables
PREFIX=	Specifies a prefix for naming the canonical variables
SINGULAR=	Specifies the singularity criterion
Control Displayed Output	
ALL	Displays all output
ANOVA	Displays univariate statistics
BCORR	Displays between correlations
BCOV	Displays between covariances
BSSCP	Displays between SSCPs
DISTANCE	Displays squared Mahalanobis distances
NOPRINT	Suppresses all displayed output
PCORR	Displays pooled correlations
PCOV	Displays pooled covariances
PSSCP	Displays pooled SSCPs
SHORT	Suppresses some displayed output
SIMPLE	Displays simple descriptive statistics
STDMEAN	Displays standardized class means
TCORR	Displays total correlations
TCOV	Displays total covariances
TSSCP	Displays total SSCPs
WCORR	Displays within correlations
WCOV	Displays within covariances
WSSCP	Displays within SSCPs

The following list provides details about these *options*.

ALL

activates all the display options.

ANOVA

displays univariate statistics for testing the hypothesis that the class means are equal in the population for each variable.

BCORR

displays between-class correlations.

BCOV

displays between-class covariances. The between-class covariance matrix equals the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes. The between-class covariances should be interpreted in comparison with the total-sample and within-class covariances, not as formal estimates of population parameters.

BSSCP

displays the between-class SSCP matrix.

DATA=*SAS-data-set*

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

If PROC HPCANDISC 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 the procedure reads the data alongside the distributed database. For more information, see the section "Processing Modes" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*) about the various execution modes and the section "Alongside-the-Database Execution" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*) about the alongside-the-database model.

DISTANCE

MAHALANOBIS

displays squared Mahalanobis distances between the group means, the F statistics, and the corresponding probabilities of greater squared Mahalanobis distances between the group means.

NCAN=*n*

specifies the number of canonical variables to be computed. The value of *n* must be less than or equal to the number of variables. If you specify NCAN=0, PROC HPCANDISC displays the canonical correlations but not the canonical coefficients, structures, or means. A negative value suppresses the canonical analysis entirely. Let v be the number of variables in the VAR statement, and let c be the number of classes. If you omit the NCAN= option, only $\min(v, c-1)$ canonical variables are generated; if you also specify an OUT= output data set, v canonical variables are generated, and the last $v-(c-1)$ canonical variables have missing values.

NOPRINT

suppresses the normal display of results. This option temporarily disables the Output Delivery System (ODS). For more information about ODS, see Chapter 20, "Using the Output Delivery System."

OUT=*SAS-data-set*

creates an output SAS data set to contain observationwise canonical variable scores. The variables in the input data set are *not* included in the output data set to avoid data duplication for large data sets; 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 HPCANDISC 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 4408.

OUTSTAT=*SAS-data-set*

creates a TYPE=CORR output SAS data set to contain various statistics, including class means, standard deviations, correlations, canonical correlations, canonical structures, canonical coefficients, and means of canonical variables for each class level.

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*.

PCORR

displays pooled within-class correlations (partial correlations based on the pooled within-class covariances).

PCOV

displays pooled within-class covariances.

PREFIX=*name*

specifies a prefix for naming the canonical variables. By default, the names are Can1, Can2, Can3, and so on. If you specify PREFIX=Abc, the components are named Abc1, Abc2, and so on. The number of characters in the prefix plus the number of digits required to designate the canonical variables should not exceed 32. The prefix is truncated if the combined length exceeds 32.

PSSCP

displays the pooled within-class corrected SSCP matrix.

SHORT

suppresses the display of canonical structures, canonical coefficients, and class means on canonical variables; only tables of canonical correlations and multivariate test statistics are displayed.

SIMPLE

displays simple descriptive statistics for the total sample and within each class.

SINGULAR=*p*

specifies the criterion for determining the singularity of the total-sample correlation matrix and the pooled within-class covariance matrix, where $0 < p < 1$. The default is SINGULAR=1E–8.

Let **S** be the total-sample correlation matrix. If the R square for predicting a quantitative variable in the VAR statement from the variables that precede it exceeds $1 - p$, then **S** is considered singular. If **S** is singular, the probability levels for the multivariate test statistics and canonical correlations are adjusted for the number of variables whose R square exceeds $1 - p$.

If **S** is considered singular and the inverse of **S** (squared Mahalanobis distances) is required, a quasi inverse is used instead. For more information, see the section "Quasi-inverse" on page 2602.

STDMEAN
displays total-sample and pooled within-class standardized class means.

TCORR
displays total-sample correlations.

TCOV
displays total-sample covariances.

TSSCP
displays the total-sample corrected SSCP matrix.

WCORR
displays within-class correlations for each class level.

WCOV
displays within-class covariances for each class level.

WSSCP
displays the within-class corrected SSCP matrix for each class level.

BY Statement

BY *variables* **;**

You can specify a BY statement in PROC HPCANDISC 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 HPCANDISC procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

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

CLASS Statement

CLASS *variable* **;**

The values of the CLASS variable define the groups for analysis. Class levels are determined by the formatted values of the CLASS variable. The CLASS variable can be numeric or character. A CLASS statement is required.

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 total number of observations is considered to be equal to the sum of the FREQ variable when the procedure determines degrees of freedom for significance probabilities.

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*).

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 HPCANDISC procedure.

You can also use the PERFORMANCE statement to control whether the HPCANDISC 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* ;

You specify the quantitative variables to include in the analysis by using a VAR statement. If you do not use a VAR statement, the analysis includes all numeric variables that are not listed in other statements.

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 a WEIGHT statement is not included, all observations that are used in the analysis are assigned a weight of 1.

The WEIGHT statement does not alter the degrees of freedom.

Details: HPCANDISC Procedure

Missing Values

If an observation has a missing value for any of the quantitative variables, it is omitted from the analysis. If an observation has a missing CLASS value but is otherwise complete, PROC HPCANDISC does not use it in computing the canonical correlations and coefficients; however, canonical variable scores are computed for that observation for the OUT= data set.

Computational Method

General Formulas

Canonical discriminant analysis is equivalent to canonical correlation analysis between the quantitative variables and a set of dummy variables coded from the CLASS variable. In the following notation, the dummy variables are denoted by $\mathbf{y}$ and the quantitative variables are denoted by $\mathbf{x}$. The total sample covariance matrix for the $\mathbf{x}$ and $\mathbf{y}$ variables is

$$\mathrm{S} = \begin{bmatrix} \mathrm{S}_{xx} & \mathrm{S}_{xy} \\ \mathrm{S}_{yx} & \mathrm{S}_{yy} \end{bmatrix}$$

When c is the number of groups, n_t is the number of observations in group t, and $\mathbf{S}_t$ is the sample covariance matrix for the $\mathbf{x}$ variables in group t, the within-class pooled covariance matrix for the $\mathbf{x}$ variables is

$$\mathrm{S}_p = \frac{1}{\sum n_t - c} \sum (n_t - 1)\mathrm{S}_t$$

The canonical correlations, ρ_i, are the square roots of the eigenvalues, λ_i, of the following matrix. The corresponding eigenvectors are $\mathbf{v}_i$.

$$\mathbf{S}_p{}^{-1/2}\mathbf{S}_{xy}\mathbf{S}_{yy}{}^{-1}\mathbf{S}_{yx}\mathbf{S}_p{}^{-1/2}$$

Let $\mathbf{V}$ be the matrix that contains the eigenvectors $\mathbf{v}_i$ that correspond to nonzero eigenvalues as columns. The raw canonical coefficients are calculated as follows:

$$\mathbf{R} = \mathbf{S}_p{}^{-1/2}\mathbf{V}$$

The pooled within-class standardized canonical coefficients are

$$\mathbf{P} = \mathrm{diag}(\mathbf{S}_p)^{1/2}\mathbf{R}$$

The total sample standardized canonical coefficients are

$$\mathbf{T} = \mathrm{diag}(\mathbf{S}_{xx})^{1/2}\mathbf{R}$$

Let $\mathbf{X}_c$ be the matrix that contains the centered $\mathbf{x}$ variables as columns. The canonical scores can be calculated by any of the following:

$$\mathbf{X}_c\,\mathbf{R}$$

$$\mathbf{X}_c\,\mathrm{diag}(\mathbf{S}_p)^{-1/2}\mathbf{P}$$

$$\mathbf{X}_c\,\mathrm{diag}(\mathbf{S}_{xx})^{-1/2}\mathbf{T}$$

For the multivariate tests based on $\mathbf{E}^{-1}\mathbf{H}$,

$$\mathbf{E} = (n-1)(\mathbf{S}_{yy} - \mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy})$$

$$\mathbf{H} = (n-1)\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}$$

where n is the total number of observations.

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 HPCANDISC 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 statement, PROC HPCANDISC 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 HPCANDISC procedure allocates one thread per CPU.

The tasks that are multithreaded by the HPCANDISC procedure are primarily defined by dividing the data processed on a single machine among the threads; that is, PROC HPCANDISC 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. These operations include the following:

- variable levelization
- formation of the crossproducts matrix
- canonical variable scoring of observations

Output Data Sets

OUT= Data Set

Many SAS procedures add the variables from the input data set when an observationwise output data set is created. The assumption of high-performance analytics procedures is that the input data sets can be large and can contain many variables. For performance reasons, the OUT= data set contains the following:

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

Having these variables and keys in the OUT= data set 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 HPCANDISC is run in distributed mode, see the section “Output Data Sets” (Chapter 2, *SAS/STAT User’s Guide: High-Performance Procedures*).

The new variables that are created for the OUT= data set contain the canonical variable scores. You determine the number of new variables by using the NCAN= option. The names of the new variables are formed as they are for the PREFIX= option. The new variables have means equal to 0 and pooled within-class variances equal to 1.

OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR data set that the CORR procedure produces but contains many results in addition to those produced by PROC CORR.

The OUTSTAT= data set is TYPE=CORR, and it contains the following variables:

- the BY variables, if any
- the CLASS variable
- _TYPE_, a character variable of length 8 that identifies the type of statistic
- _NAME_, a character variable of length 32 that identifies the row of the matrix or the name of the canonical variable
- the quantitative variables (those in the VAR statement, or if there is no VAR statement, all numeric variables not listed in any other statement)

The observations, as identified by the variable _TYPE_, have the following _TYPE_ values:

TYPE	**Contents**
N	number of observations for the total sample (CLASS variable missing) and within each class (CLASS variable present)
SUMWGT	sum of weights for the total sample (CLASS variable missing) and within each class (CLASS variable present) if a WEIGHT statement is specified
MEAN	means for the total sample (CLASS variable missing) and within each class (CLASS variable present)
STDMEAN	total-standardized class means
PSTDMEAN	pooled within-class standardized class means
STD	standard deviations for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PSTD	pooled within-class standard deviations
BSTD	between-class standard deviations
RSQUARED	univariate R squares

The following kinds of observations are identified by the combination of the variables _TYPE_ and _NAME_. When the _TYPE_ variable has one of the following values, the _NAME_ variable identifies the row of the matrix:

TYPE	**Contents**
CSSCP	corrected SSCP matrix for the total sample (CLASS variable missing) and within each class (CLASS variable present)

PSSCP	pooled within-class corrected SSCP matrix
BSSCP	between-class SSCP matrix
COV	covariance matrix for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PCOV	pooled within-class covariance matrix
BCOV	between-class covariance matrix
CORR	correlation matrix for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PCORR	pooled within-class correlation matrix
BCORR	between-class correlation matrix

When the _TYPE_ variable has one of the following values, the _NAME_ variable identifies the canonical variable:

TYPE	**Contents**
CANCORR	canonical correlations
STRUCTUR	canonical structure
BSTRUCT	between canonical structure
PSTRUCT	pooled within-class canonical structure
SCORE	total-sample standardized canonical coefficients
PSCORE	pooled within-class standardized canonical coefficients
RAWSCORE	raw canonical coefficients
CANMEAN	means of the canonical variables for each class

You can use this data set in PROC SCORE to get scores on the canonical variables for new data by using one of the following forms:

```
* The CLASS variable C is numeric;
proc score data=NewData score=Coef(where=(c = .  )) out=Scores;
run;

* The CLASS variable C is character;
proc score data=NewData score=Coef(where=(c = ' ')) out=Scores;
run;
```

The WHERE clause excludes the within-class means and standard deviations. PROC SCORE standardizes the new data by subtracting the original variable means that are stored in the _TYPE_=’MEAN’ observations and dividing by the original variable standard deviations from the _TYPE_=’STD’ observations. Then PROC SCORE multiplies the standardized variables by the coefficients from the _TYPE_=’SCORE’ observations to get the canonical scores.

Displayed Output

By default, the HPCANDISC procedure begins by displaying the output along with the following:

- The "Performance Information" table, which 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.
- The "Data Access Information" table, which 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.
- Summary information about the variables in the analysis that displays the total sample size, the number of quantitative variables, the number of class levels, and the number of degrees of freedom
- The "Number of Observations" table, which 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.
- The "Class Level Information" table, which displays, for each level of the classification variable, the frequency sum, weight sum, and proportion of the total sample

The optional output from PROC HPCANDISC includes the following:

- Within-class SSCP matrices for each group
- Pooled within-class SSCP matrix
- Between-class SSCP matrix
- Total-sample SSCP matrix
- Within-class covariance matrices for each group
- Pooled within-class covariance matrix
- Between-class covariance matrix, equal to the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes
- Total-sample covariance matrix
- Within-class correlation coefficients and $\mathrm{Pr} > |r|$ to test the hypothesis that the within-class population correlation coefficients are zero
- Pooled within-class correlation coefficients and $\mathrm{Pr} > |r|$ to test the hypothesis that the partial population correlation coefficients are zero
- Between-class correlation coefficients and $\mathrm{Pr} > |r|$ to test the hypothesis that the between-class population correlation coefficients are zero

- Total-sample correlation coefficients and $\Pr > |r|$ to test the hypothesis that the total population correlation coefficients are zero

- Simple statistics, including N (the number of observations), sum, mean, variance, and standard deviation for the total sample and within each class

- Total-sample standardized class means, obtained by subtracting the grand mean from each class mean and dividing by the total sample standard deviation

- Pooled within-class standardized class means, obtained by subtracting the grand mean from each class mean and dividing by the pooled within-class standard deviation

- Pairwise squared distances between groups

- Univariate test statistics, including total-sample standard deviations, pooled within-class standard deviations, between-class standard deviations, R square, $R^2/(1-R^2)$, F, and $\Pr > F$ (univariate F values and probability levels for one-way analyses of variance)

- The "Timing" table, which displays the elapsed time for each main task of the procedure, if you specify the DETAILS option in the PERFORMANCE statement

By default, PROC HPCANDISC displays these statistics:

- Multivariate statistics and F approximations, including Wilks' lambda, Pillai's trace, Hotelling-Lawley trace, and Roy's greatest root with F approximations, numerator and denominator degrees of freedom (Num DF and Den DF), and probability values ($\Pr > F$). Each of these four multivariate statistics tests the hypothesis that the class means are equal in the population. For more information, see the section "Multivariate Tests" on page 96.

- Canonical correlations

- Adjusted canonical correlations (Lawley 1959). These are asymptotically less biased than the raw correlations and can be negative. The adjusted canonical correlations might not be computable and are displayed as missing values if two canonical correlations are nearly equal or if some are close to zero. A missing value is also displayed if an adjusted canonical correlation is larger than a previous adjusted canonical correlation.

- Approximate standard error of the canonical correlations

- Squared canonical correlations

- Eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$. Each eigenvalue is equal to $\rho^2/(1-\rho^2)$, where ρ^2 is the corresponding squared canonical correlation and can be interpreted as the ratio of between-class variation to pooled within-class variation for the corresponding canonical variable. The table includes eigenvalues, differences between successive eigenvalues, the proportion of the sum of the eigenvalues, and the cumulative proportion.

- Likelihood ratio for the hypothesis that the current canonical correlation and all smaller ones are zero in the population. The likelihood ratio for the hypothesis that all canonical correlations equal zero is Wilks' lambda.

- Approximate F statistic based on Rao's approximation to the distribution of the likelihood ratio (Rao 1973, p. 556; Kshirsagar 1972, p. 326)
- Numerator degrees of freedom (Num DF), denominator degrees of freedom (Den DF), and $\Pr > F$, the probability level associated with the F statistic

You can suppress the following statistics by specifying the SHORT option:

- Total canonical structure, giving total-sample correlations between the canonical variables and the original variables
- Between canonical structure, giving between-class correlations between the canonical variables and the original variables
- Pooled within canonical structure, giving pooled within-class correlations between the canonical variables and the original variables
- Total-sample standardized canonical coefficients, standardized to give canonical variables that have zero mean and unit pooled within-class variance when applied to the total-sample standardized variables
- Pooled within-class standardized canonical coefficients, standardized to give canonical variables that have zero mean and unit pooled within-class variance when applied to the pooled within-class standardized variables
- Raw canonical coefficients, standardized to give canonical variables that have zero mean and unit pooled within-class variance when applied to the centered variables
- Class means on the canonical variables

ODS Table Names

PROC HPCANDISC 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 54.2. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 54.2 ODS Tables Produced by PROC HPCANDISC

Table Name	Description	Required Statement / Option
ANOVA	Univariate statistics	ANOVA
AveRSquare	Average R square	ANOVA
BCorr	Between-class correlations	BCORR
BCov	Between-class covariances	BCOV
BSSCP	Between-class SSCP matrix	BSSCP
BStruc	Between canonical structure	Default
CanCorr	Canonical correlations	Default
CanonicalMeans	Class means on canonical variables	Default

Table 54.2 *continued*

Table Name	Description	Required Statement / Option
ClassLevels	Class level information	Default
Counts	Number of observations, variables, class levels, *df*	Default
DataAccessInfo	Information about modes of data access	Default
Dist	Squared distances	DISTANCE
DistFValues	*F* statistics based on squared distances	DISTANCE
DistProb	Probabilities for *F* statistics from squared distances	DISTANCE
MultStat	MANOVA	Default
NObs	Number of observations	Default
PCoef	Pooled standard canonical coefficients	Default
PCorr	Pooled within-class correlations	PCORR
PCov	Pooled within-class covariances	PCOV
PerformanceInfo	Information about the high-performance computing environment	Default
PSSCP	Pooled within-class SSCP matrix	PSSCP
PStdMeans	Pooled standardized class means	STDMEAN
PStruc	Pooled within canonical structure	Default
RCoef	Raw canonical coefficients	Default
SimpleStatistics	Simple statistics	SIMPLE
TCoef	Total-sample standard canonical coefficients	Default
TCorr	Total-sample correlations	TCORR
TCov	Total-sample covariances	TCOV
Timing	Absolute and relative times for tasks performed by the procedure	PERFORMANCE DETAILS
TSSCP	Total-sample SSCP matrix	TSSCP
TStdMeans	Total standardized class means	STDMEAN
TStruc	Total canonical structure	Default
WCorr	Within-class correlations	WCORR
WCov	Within-class covariances	WCOV
WSSCP	Within-class SSCP matrices	WSSCP

Examples: HPCANDISC Procedure

Example 54.1: Analyzing Iris Data with PROC HPCANDISC

The iris data that were published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters in 50 iris specimens from each of three species: *Iris setosa*, *I. versicolor*, and *I. virginica*. The iris data set is available from the Sashelp library.

This example is a canonical discriminant analysis that creates an output data set that contains scores on the canonical variables and plots the canonical variables. The ID statement is specified to add the variable Species from the input data set to the output data set.

The following statements produce Output 54.1.1 through Output 54.1.6:

```
title 'Fisher (1936) Iris Data';

proc hpcandisc data=sashelp.iris out=outcan distance anova;
   id Species;
   class Species;
   var SepalLength SepalWidth PetalLength PetalWidth;
run;
```

Output 54.1.1 displays performance information, data access information, and summary information about the observations and the classes in the data set.

Output 54.1.1 Iris Data: Performance, Data Access, and Summary Information

Fisher (1936) Iris Data

The HPCANDISC Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	16

Data Access Information

Data	Engine	Role	Path
SASHELP.IRIS	V9	Input	On Client
WORK.OUTCAN	V9	Output	On Client

Total Sample Size	150	**DF Total**	149
Variables	4	**DF Within Classes**	147
Class Levels	3	**DF Between Classes**	2

Number of Observations Read	150
Number of Observations Used	150

Class Level Information

Species	Frequency	Weight	Proportion
Setosa	50	50.00000	0.33333
Versicolor	50	50.00000	0.33333
Virginica	50	50.00000	0.33333

Output 54.1.2 shows results from the DISTANCE option in the PROC HPCANDISC statement, which display squared Mahalanobis distances between class means.

Output 54.1.2 Iris Data: Squared Mahalanobis Distances and Distance Statistics

Fisher (1936) Iris Data

The HPCANDISC Procedure

Squared Distance to Species

From Species	Setosa	Versicolor	Virginica
Setosa	0	89.86419	179.38471
Versicolor	89.86419	0	17.20107
Virginica	179.38471	17.20107	0

F Statistics, Num DF=4, Den DF=144 for Squared Distance to Species

From Species	Setosa	Versicolor	Virginica
Setosa	0	550.18889	1098.27375
Versicolor	550.18889	0	105.31265
Virginica	1098.27375	105.31265	0

Prob > Mahalanobis Distance for Squared Distance to Species

From Species	Setosa	Versicolor	Virginica
Setosa	1.0000	<.0001	<.0001
Versicolor	<.0001	1.0000	<.0001
Virginica	<.0001	<.0001	1.0000

Output 54.1.3 displays univariate and multivariate statistics. The ANOVA option uses univariate statistics to test the hypothesis that the class means are equal. The resulting R-square values range from 0.4008 for SepalWidth to 0.9414 for PetalLength, and each variable is significant at the 0.0001 level. The multivariate test for differences between the class levels (which is displayed by default) is also significant at the 0.0001 level; you would expect this from the highly significant univariate test results.

Output 54.1.3 Iris Data: Univariate and Multivariate Statistics

Fisher (1936) Iris Data

The HPCANDISC Procedure

Univariate Test Statistics

F Statistics, Num DF=2, Den DF=147

Variable	Label	Total Standard Deviation	Pooled Standard Deviation	Between Standard Deviation	R-Square	R-Square / (1-Rsq)	F Value	Pr > F
SepalLength	Sepal Length (mm)	8.28066	5.14789	7.95061	0.6187	1.6226	119.26	<.0001
SepalWidth	Sepal Width (mm)	4.35866	3.39688	3.36822	0.4008	0.6688	49.16	<.0001
PetalLength	Petal Length (mm)	17.65298	4.30334	20.90700	0.9414	16.0566	1180.16	<.0001
PetalWidth	Petal Width (mm)	7.62238	2.04650	8.96735	0.9289	13.0613	960.01	<.0001

Average R-Square	
Unweighted	0.7224358
Weighted by Variance	0.8689444

Multivariate Statistics and F Approximations

S=2 M=0.5 N=71

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.023439	199.15	8	288	<.0001
Pillai's Trace	1.191899	53.47	8	290	<.0001
Hotelling-Lawley Trace	32.477320	582.20	8	203.4	<.0001
Roy's Greatest Root	32.191929	1166.96	4	145	<.0001

NOTE: F Statistic for Roy's Greatest Root is an upper bound.

NOTE: F Statistic for Wilks' Lambda is exact.

Output 54.1.4 displays canonical correlations and eigenvalues. The R square between Can1 and the CLASS variable, 0.969872, is much larger than the corresponding R square for Can2, 0.222027.

Output 54.1.4 Iris Data: Canonical Correlations and Eigenvalues

Fisher (1936) Iris Data

The HPCANDISC Procedure

					Eigenvalues of Inv(E)*H = CanRsq/(1-CanRsq)			
		Adjusted	Approximate	Squared				
	Canonical	Canonical	Standard	Canonical				
	Correlation	Correlation	Error	Correlation	Eigenvalue	Difference	Proportion	Cumulative
1	0.984821	0.984508	0.002468	0.969872	32.1919	31.9065	0.9912	0.9912
2	0.471197	0.461445	0.063734	0.222027	0.2854		0.0088	1.0000

Test of H0: The canonical correlations in the current row and all that follow are zero					
	Likelihood	Approximate			
	Ratio	F Value	Num DF	Den DF	Pr > F
1	0.02343863	199.15	8	288	<.0001
2	0.77797337	13.79	3	145	<.0001

Output 54.1.5 displays correlations between canonical and original variables.

Output 54.1.5 Iris Data: Correlations between Canonical and Original Variables

Fisher (1936) Iris Data

The HPCANDISC Procedure

Total Canonical Structure			
Variable	Label	Can1	Can2
SepalLength	Sepal Length (mm)	0.79189	0.21759
SepalWidth	Sepal Width (mm)	-0.53076	0.75799
PetalLength	Petal Length (mm)	0.98495	0.04604
PetalWidth	Petal Width (mm)	0.97281	0.22290

Between Canonical Structure			
Variable	Label	Can1	Can2
SepalLength	Sepal Length (mm)	0.99147	0.13035
SepalWidth	Sepal Width (mm)	-0.82566	0.56417
PetalLength	Petal Length (mm)	0.99975	0.02236
PetalWidth	Petal Width (mm)	0.99404	0.10898

Pooled Within Canonical Structure			
Variable	Label	Can1	Can2
SepalLength	Sepal Length (mm)	0.22260	0.31081
SepalWidth	Sepal Width (mm)	-0.11901	0.86368
PetalLength	Petal Length (mm)	0.70607	0.16770
PetalWidth	Petal Width (mm)	0.63318	0.73724

Output 54.1.6 displays canonical coefficients. The raw canonical coefficients for the first canonical variable, Can1, show that the class levels differ most widely on the linear combination of the centered variables: $-0.0829378 \times \text{SepalLength} - 0.153447 \times \text{SepalWidth} + 0.220121 \times \text{PetalLength} + 0.281046 \times \text{PetalWidth}$.

Output 54.1.6 Iris Data: Canonical Coefficients

Fisher (1936) Iris Data

The HPCANDISC Procedure

Total-Sample Standardized Canonical Coefficients

Variable	Label	Can1	Can2
SepalLength	Sepal Length (mm)	-0.68678	0.01996
SepalWidth	Sepal Width (mm)	-0.66883	0.94344
PetalLength	Petal Length (mm)	3.88580	-1.64512
PetalWidth	Petal Width (mm)	2.14224	2.16414

Pooled Within-Class Standardized Canonical Coefficients

Variable	Label	Can1	Can2
SepalLength	Sepal Length (mm)	-0.42695	0.01241
SepalWidth	Sepal Width (mm)	-0.52124	0.73526
PetalLength	Petal Length (mm)	0.94726	-0.40104
PetalWidth	Petal Width (mm)	0.57516	0.58104

Raw Canonical Coefficients

Variable	Label	Can1	Can2
SepalLength	Sepal Length (mm)	-0.08294	0.00241
SepalWidth	Sepal Width (mm)	-0.15345	0.21645
PetalLength	Petal Length (mm)	0.22012	-0.09319
PetalWidth	Petal Width (mm)	0.28105	0.28392

Output 54.1.7 displays class means on canonical variables.

Output 54.1.7 Iris Data: Canonical Means

Class Means on Canonical Variables

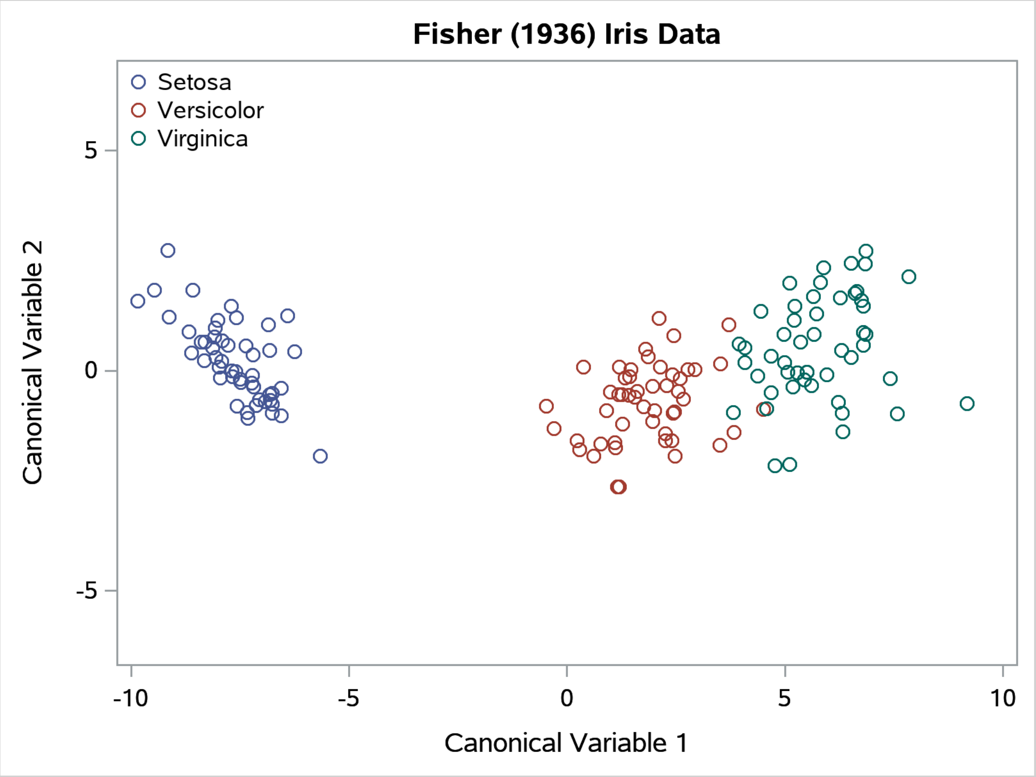
Species	Can1	Can2
Setosa	-7.60760	0.21513
Versicolor	1.82505	-0.72790
Virginica	5.78255	0.51277

The TEMPLATE and SGRENDER procedures are used to create a plot of the first two canonical variables. The following statements produce Output 54.1.8:

```
proc template;
   define statgraph scatter;
      begingraph;
         entrytitle 'Fisher (1936) Iris Data';
         layout overlayequated / equatetype=fit
```

```
               xaxisopts=(label='Canonical Variable 1')
               yaxisopts=(label='Canonical Variable 2');
               scatterplot x=Can1 y=Can2 / group=species name='iris';
               layout gridded / autoalign=(topleft);
                  discretelegend 'iris' / border=false opaque=false;
               endlayout;
            endlayout;
         endgraph;
      end;
   run;

   proc sgrender data=outcan template=scatter;
   run;
```

Output 54.1.8 Iris Data: Plot of First Two Canonical Variables



The plot of canonical variables in Output 54.1.8 shows that of the two canonical variables, Can1 has more discriminatory power.

Example 54.2: Performing Canonical Discriminant Analysis in Single-Machine and Distributed Modes

PROC HPCANDISC shows its real power when the computation is conducted using multiple threads or in a distributed environment.

This example shows how you can run PROC HPCANDISC 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 HPCANDISC. The following DATA step generates the data:

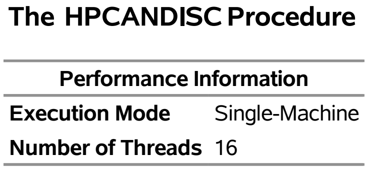
```
data ex2Data;
   drop i j n n1 n2 n3 n4;

   n  = 5000000;
   n1 = n*0.1;
   n2 = n*0.25;
   n3 = n*0.45;
   n4 = n*0.7;

   array x{100};

   do i=1 to n;
      do j=1 to dim(x);
         x{j} = ranuni(1);
      end;

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

      output;
   end;
run;
```

The following statements use PROC HPCANDISC to perform a canonical discriminant analysis and to output various statistics to the stats data set (OUTSTAT= stats).

```
proc hpcandisc data=ex2Data outstat=stats;
   var x:;
   class z;
   performance details;
run;
```

Output 54.2.1 shows the "Performance Information" table. This table shows that the HPCANDISC 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 54.2.1 Performance Information in Single-Machine Mode



The HPCANDISC Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	16

Output 54.2.2 shows timing information for the PROC HPCANDISC 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, levelizing, and processing the data.

Output 54.2.2 Timing in Single-Machine Mode

Procedure Task Timing

Task	Seconds	Percent
Reading, Levelizing, and Processing Data	40.60	99.79%
Computing SSCP and Covariance Matrices	0.00	0.00%
Performing Canonical Analysis	0.03	0.06%
Producing Output Statistics Data Set	0.06	0.14%

To switch to running PROC HPCANDISC 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*).

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 hpcandisc data=ex2Data outstat=stats;
   var x:;
   class z;
   performance details nodes = 4
               host="&GRIDHOST" install="&GRIDINSTALLLOC";
run;
```

The execution mode in the “Performance Information” table shown in Output 54.2.3 indicates that the calculations were performed in a distributed environment that uses four nodes, each of which uses 32 threads.

Output 54.2.3 Performance Information in Distributed Mode

Performance Information

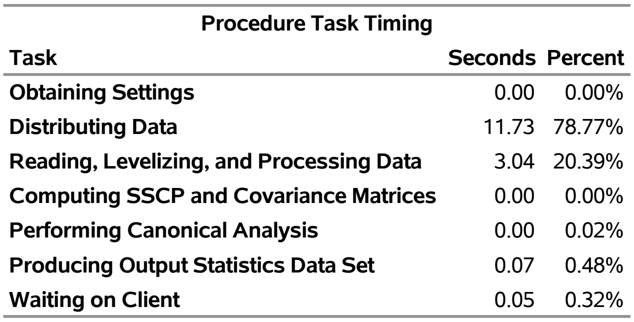
Host Node	<< your grid host >>
Install Location	<< your grid install location >>
Execution Mode	Distributed
Number of Compute Nodes	4
Number of Threads per Node	32

Another indication of distributed execution is the following message issued by all high-performance analytics procedures in the SAS log:

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

Output 54.2.4 shows timing information for this distributed run of the HPCANDISC procedure. In contrast to the single-machine mode (where reading, levelizing, and processing the data dominated the time spent), the majority of time in the distributed mode run is spent distributing the data.

Output 54.2.4 Timing in Distributed Mode



```
                    Procedure Task Timing
Task                                          Seconds  Percent
Obtaining Settings                               0.00    0.00%
Distributing Data                               11.73   78.77%
Reading, Levelizing, and Processing Data         3.04   20.39%
Computing SSCP and Covariance Matrices           0.00    0.00%
Performing Canonical Analysis                    0.00    0.02%
Producing Output Statistics Data Set             0.07    0.48%
Waiting on Client                                0.05    0.32%
```

References

Fisher, R. A. (1936). "The Use of Multiple Measurements in Taxonomic Problems." *Annals of Eugenics* 7:179–188.

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

Lawley, D. N. (1959). "Tests of Significance in Canonical Analysis." *Biometrika* 46:59–66.

Puranen, J. (1917). "Fish Catch data set (1917)." Journal of Statistics Education Data Archive.

Rao, C. R. (1973). *Linear Statistical Inference and Its Applications*. 2nd ed. New York: John Wiley & Sons.

Subject Index

Syntax Index