Title

stata.com

discrim lda postestimation — Postestimation tools for discrim lda

Description

The following postestimation commands are of special interest after `discrim lda`:

Command	Description
`estat anova`	ANOVA summaries table
`estat canontest`	tests of the canonical discriminant functions
`estat classfunctions`	classification functions
`estat classtable`	classification table
`estat correlations`	correlation matrices and p-values
`estat covariance`	covariance matrices
`estat errorrate`	classification error-rate estimation
`estat grdistances`	Mahalanobis and generalized squared distances between the group means
`estat grmeans`	group means and variously standardized or transformed means
`estat grsummarize`	group summaries
`estat list`	classification listing
`estat loadings`	canonical discriminant-function coefficients (loadings)
`estat manova`	MANOVA table
`estat structure`	canonical structure matrix
`estat summarize`	estimation sample summary
`loadingplot`	plot standardized discriminant-function loadings
`scoreplot`	plot discriminant-function scores
`screeplot`	plot eigenvalues

The following standard postestimation commands are also available:

Command	Description
* `estimates`	cataloging estimation results
`predict`	group classification and posterior probabilities

*All `estimates` subcommands except `table` and `stats` are available; see [R] **estimates**.

Special-interest postestimation commands

`estat anova` presents a table summarizing the one-way ANOVAs for each variable in the discriminant analysis.

`estat canontest` presents tests of the canonical discriminant functions. Presented are the canonical correlations, eigenvalues, proportion and cumulative proportion of variance, and likelihood-ratio tests for the number of nonzero eigenvalues.

`estat classfunctions` displays the classification functions.

`estat correlations` displays the pooled within-group correlation matrix, between-groups correlation matrix, total-sample correlation matrix, and/or the individual group correlation matrices. Two-tailed p-values for the correlations may also be requested.

`estat covariance` displays the pooled within-group covariance matrix, between-groups covariance matrix, total-sample covariance matrix, and/or the individual group covariance matrices.

`estat grdistances` provides Mahalanobis squared distances between the group means along with the associated F statistics and significance levels. Also available are generalized squared distances.

`estat grmeans` provides group means, total-sample standardized group means, pooled within-group standardized means, and canonical functions evaluated at the group means.

`estat loadings` present the canonical discriminant-function coefficients (loadings). Unstandardized, pooled within-class standardized, and total-sample standardized coefficients are available.

`estat manova` presents the MANOVA table associated with the discriminant analysis.

`estat structure` presents the canonical structure matrix.

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic options*]

`predict` [*type*] { *stub*`*` | *newvarlist* } [*if*] [*in*] [, *statistic options*]

statistic	Description
Main	
`classification`	group membership classification; the default when one variable is specified and `group()` is not specified
`pr`	probability of group membership; the default when `group()` is specified or when multiple variables are specified
`mahalanobis`	Mahalanobis squared distance between observations and groups
`dscore`	discriminant function score
`clscore`	group classification function score
* `looclass`	leave-one-out group membership classification; may be used only when one new variable is specified
* `loopr`	leave-one-out probability of group membership
* `loomahal`	leave-one-out Mahalanobis squared distance between observations and groups

options	Description
Main	
`group(`*group*`)`	the group for which the statistic is to be calculated
Options	
`priors(`*priors*`)`	group prior probabilities; defaults to `e(grouppriors)`
`ties(`*ties*`)`	how ties in classification are to be handled; defaults to `e(ties)`

priors	Description
`equal`	equal prior probabilities
`proportional`	group-size-proportional prior probabilities
matname	row or column vector containing the group prior probabilities
matrix_exp	matrix expression providing a row or column vector of the group prior probabilities

ties	Description
`missing`	ties in group classification produce missing values
`random`	ties in group classification are broken randomly
`first`	ties in group classification are set to the first tied group

You specify one new variable with `classification` or `looclass`; either one or `e(N_groups)` new variables with `pr`, `loopr`, `mahalanobis`, `loomahal`, or `clscore`; and one to `e(f)` new variables with `dscore`.

Unstarred statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample. Starred statistics are calculated only for the estimation sample, even when `if e(sample)` is not specified.

`group()` is not allowed with `classification`, `dscore`, or `looclass`.

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`classification`, the default, calculates the group classification. Only one new variable may be specified.

`pr` calculates group membership posterior probabilities. If you specify the `group()` option, specify one new variable. Otherwise, you must specify `e(N_groups)` new variables.

`mahalanobis` calculates the squared Mahalanobis distance between the observations and group means. If you specify the `group()` option, specify one new variable. Otherwise, you must specify `e(N_groups)` new variables.

`dscore` produces the discriminant function score. Specify as many variables as leading discriminant functions that you wish to score. No more than `e(f)` variables may be specified.

`clscore` produces the group classification function score. If you specify the `group()` option, specify one new variable. Otherwise, you must specify `e(N_groups)` new variables.

`looclass` calculates the leave-one-out group classifications. Only one new variable may be specified. Leave-one-out calculations are restricted to `e(sample)` observations.

`loopr` calculates the leave-one-out group membership posterior probabilities. If you specify the `group()` option, specify one new variable. Otherwise, you must specify `e(N_groups)` new variables. Leave-one-out calculations are restricted to `e(sample)` observations.

`loomahal` calculates the leave-one-out squared Mahalanobis distance between the observations and group means. If you specify the `group()` option, specify one new variable. Otherwise, you must specify `e(N_groups)` new variables. Leave-one-out calculations are restricted to `e(sample)` observations.

`group(`*group*`)` specifies the group for which the statistic is to be calculated and can be specified using

`#1`, `#2`, ..., where `#1` means the first category of the `e(groupvar)` variable, `#2` the second category, etc.;

the values of the `e(groupvar)` variable; or

the value labels of the `e(groupvar)` variable if they exist.

`group()` is not allowed with `classification`, `dscore`, or `looclass`.

Options

`priors(`*priors*`)` specifies the prior probabilities for group membership. If `priors()` is not specified, `e(grouppriors)` is used. The following *priors* are allowed:

`priors(equal)` specifies equal prior probabilities.

`priors(proportional)` specifies group-size-proportional prior probabilities.

`priors(`*matname*`)` specifies a row or column vector containing the group prior probabilities.

`priors(`*matrix_exp*`)` specifies a matrix expression providing a row or column vector of the group prior probabilities.

`ties(`*ties*`)` specifies how ties in group classification will be handled. If `ties()` is not specified, `e(ties)` is used. The following *ties* are allowed:

`ties(missing)` specifies that ties in group classification produce missing values.

`ties(random)` specifies that ties in group classification are broken randomly.

`ties(first)` specifies that ties in group classification are set to the first tied group.

Syntax for estat

ANOVA summaries table

```
estat anova
```

Tests of the canonical discriminant functions

```
estat canontest
```

Classification functions

```
estat classfunctions [ , classfunctions_options ]
```

Correlation matrices and p-values

```
estat correlations [ , correlations_options ]
```

Covariance matrices

```
estat covariance [ , covariance_options ]
```

Mahalanobis and generalized squared distances between the group means

```
estat grdistances [ , grdistances_options ]
```

Group means and variously standardized or transformed means

```
estat grmeans [ , grmeans_options ]
```

Canonical discriminant-function coefficients (loadings)

```
estat loadings [ , loadings_options ]
```

MANOVA table

```
estat manova
```

Canonical structure matrix

```
estat structure [ , format(%fmt) ]
```

classfunctions_options	Description
Main	
adjustequal	adjust the constant even when priors are equal
format(%*fmt*)	numeric display format; default is %9.0g
Options	
priors(*priors*)	group prior probabilities; defaults to e(grouppriors)
nopriors	suppress display of prior probabilities

correlations_options	Description
Main	
within	display pooled within-group correlation matrix; the default
between	display between-groups correlation matrix
total	display total-sample correlation matrix
groups	display the correlation matrix for each group
all	display all the above
p	display two-sided p-values for requested correlations
format(%*fmt*)	numeric display format; default is %9.0g
nohalf	display full matrix even if symmetric

covariance_options	Description
Main	
`within`	display pooled within-group covariance matrix; the default
`between`	display between-groups covariance matrix
`total`	display total-sample covariance matrix
`groups`	display the covariance matrix for each group
`all`	display all the above
`format(`*%fmt*`)`	numeric display format; default is `%9.0g`
`nohalf`	display full matrix even if symmetric

grdistances_options	Description
Main	
`mahalanobis[ (f p) ]`	display Mahalanobis squared distances between group means; the default
`generalized`	display generalized Mahalanobis squared distances between group means
`all`	equivalent to `mahalanobis(f p) generalized`
`format(`*%fmt*`)`	numeric display format; default is `%9.0g`
Options	
`priors(`*priors*`)`	group prior probabilities; defaults to `e(grouppriors)`

grmeans_options	Description
Main	
`raw`	display untransformed and unstandardized group means
`totalstd`	display total-sample standardized group means
`withinstd`	display pooled within-group standardized group means
`canonical`	display canonical functions evaluated at group means
`all`	display all the mean tables

loadings_options	Description
Main	
`standardized`	display pooled within-group standardized canonical discriminant function coefficients; the default
`totalstandardized`	display the total-sample standardized canonical discriminant function coefficients
`unstandardized`	display unstandardized canonical discriminant function coefficients
`all`	display all the above
`format(`*%fmt*`)`	numeric display format; default is `%9.0g`

Menu for estat

Statistics > Postestimation > Reports and statistics

Options for estat

Options for `estat` are presented under the following headings:

Options for estat classfunctions

Main

`adjustequal` specifies that the constant term in the classification function be adjusted for prior probabilities even though the priors are equal. By default, equal prior probabilities are not used in adjusting the constant term. `adjustequal` has no effect with unequal prior probabilities.

`format(%`*fmt*`)` specifies the matrix display format. The default is `format(%9.0g)`.

Options

`priors(`*priors*`)` specifies the group prior probabilities. The prior probabilities affect the constant term in the classification function. By default, *priors* is determined from `e(grouppriors)`. See *Options for predict* for the *priors* specification. By common convention, when there are equal prior probabilities the adjustment of the constant term is not performed. See `adjustequal` to override this convention.

`nopriors` specifies that the prior probabilities not be displayed. By default, the prior probabilities used in determining the constant in the classification functions are displayed as the last row in the classification functions table.

Options for estat correlations

Main

`within` specifies that the pooled within-group correlation matrix be displayed. This is the default.

`between` specifies that the between-groups correlation matrix be displayed.

`total` specifies that the total-sample correlation matrix be displayed.

`groups` specifies that the correlation matrix for each group be displayed.

`all` is the same as specifying `within`, `between`, `total`, and `groups`.

`p` specifies that two-sided p-values be computed and displayed for the requested correlations.

`format(%`*fmt*`)` specifies the matrix display format. The default is `format(%8.5f)`.

`nohalf` specifies that, even though the matrix is symmetric, the full matrix be printed. The default is to print only the lower triangle.

Options for estat covariance

Main

`within` specifies that the pooled within-group covariance matrix be displayed. This is the default.

`between` specifies that the between-groups covariance matrix be displayed.

`total` specifies that the total-sample covariance matrix be displayed.

`groups` specifies that the covariance matrix for each group be displayed.

`all` is the same as specifying `within`, `between`, `total`, and `groups`.

`format(%`*fmt*`)` specifies the matrix display format. The default is `format(%9.0g)`.

`nohalf` specifies that, even though the matrix is symmetric, the full matrix be printed. The default is to print only the lower triangle.

Options for estat grdistances

Main

`mahalanobis`[`(f p)`] specifies that a table of Mahalanobis squared distances between group means be presented. `mahalanobis(f)` adds F tests for each displayed distance and `mahalanobis(p)` adds the associated p-values. `mahalanobis(f p)` adds both. The default is `mahalanobis`.

`generalized` specifies that a table of generalized Mahalanobis squared distances between group means be presented. `generalized` starts with what is produced by the `mahalanobis` option and adds a term accounting for prior probabilities. Prior probabilities are provided with the `priors()` option, or if `priors()` is not specified, by the values in `e(grouppriors)`. By common convention, if prior probabilities are equal across the groups, the prior probability term is omitted and the results from `generalized` will equal those from `mahalanobis`.

`all` is equivalent to specifying `mahalanobis(f p)` and `generalized`.

`format(%`*fmt*`)` specifies the matrix display format. The default is `format(%9.0g)`.

Options

`priors(`*priors*`)` specifies the group prior probabilities and affects only the output of the `generalized` option. By default, *priors* is determined from `e(grouppriors)`. See *Options for predict* for the *priors* specification.

Options for estat grmeans

Main

`raw`, the default, displays a table of group means.

`totalstd` specifies that a table of total-sample standardized group means be presented.

`withinstd` specifies that a table of pooled within-group standardized group means be presented.

`canonical` specifies that a table of the unstandardized canonical discriminant functions evaluated at the group means be presented.

`all` is equivalent to specifying `raw`, `totalstd`, `withinstd`, and `canonical`.

Options for estat loadings

Main

`standardized` specifies that the pooled within-group standardized canonical discriminant function coefficients be presented. This is the default.

`totalstandardized` specifies that the total-sample standardized canonical discriminant function coefficients be presented.

`unstandardized` specifies that the unstandardized canonical discriminant function coefficients be presented.

`all` is equivalent to specifying `standardized`, `totalstandardized`, and `unstandardized`.

`format(%`*fmt*`)` specifies the matrix display format. The default is `format(%9.0g)`.

Option for estat structure

Main

`format(%`*fmt*`)` specifies the matrix display format. The default is `format(%9.0g)`.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Classification tables, error rates, and listings

After `discrim`, including `discrim lda`, you can obtain classification tables, error-rate estimates, and listings; see [MV] **discrim estat**.

▷ Example 1: Predictive linear discriminant analysis

Example 1 of [MV] **manova** introduces the apple tree rootstock data from Andrews and Herzberg (1985, 357–360) and used in Rencher and Christensen (2012, 184). Descriptive linear discriminant analysis is often used after a multivariate analysis of variance (MANOVA) to explore the differences between groups found to be significantly different in the MANOVA.

We first examine the predictive aspects of the linear discriminant model on these data by examining classification tables, error-rate estimate tables, and classification listings.

To illustrate the ability of `discrim lda` and the postestimation commands of handling unequal prior probabilities, we perform our LDA using prior probabilities of 0.2 for the first four rootstock groups and 0.1 for the last two rootstock groups.

```
. use http://www.stata-press.com/data/r13/rootstock
(Table 6.2 Rootstock Data, Rencher and Christensen (2012))
. discrim lda y1 y2 y3 y4, group(rootstock) priors(.2, .2, .2, .2, .1, .1)
Linear discriminant analysis
Resubstitution classification summary
```

Key
Number Percent

True rootstock	Classified 1	2	3	4	5	6	Total
1	7	0	0	1	0	0	8
	87.50	0.00	0.00	12.50	0.00	0.00	100.00
2	0	4	2	1	1	0	8
	0.00	50.00	25.00	12.50	12.50	0.00	100.00
3	0	1	6	1	0	0	8
	0.00	12.50	75.00	12.50	0.00	0.00	100.00
4	3	0	1	4	0	0	8
	37.50	0.00	12.50	50.00	0.00	0.00	100.00
5	0	3	2	0	2	1	8
	0.00	37.50	25.00	0.00	25.00	12.50	100.00
6	3	0	0	0	2	3	8
	37.50	0.00	0.00	0.00	25.00	37.50	100.00
Total	13	8	11	7	5	4	48
	27.08	16.67	22.92	14.58	10.42	8.33	100.00
Priors	0.2000	0.2000	0.2000	0.2000	0.1000	0.1000	

The prior probabilities are reported at the bottom of the table. The classification results are based, in part, on the selection of prior probabilities.

With only 8 observations per rootstock and six rootstock groups, we have small cell counts in our table, with many zero cell counts. Because resubstitution classification tables give an overly optimistic view of classification ability, we use the `estat classtable` command to request a leave-one-out (LOO) classification table and request the reporting of average posterior probabilities in place of percentages.

```
. estat classtable, probabilities loo
Leave-one-out average-posterior-probabilities classification table
```

Key
Number Average posterior probability

True rootstock	LOO Classified 1	2	3	4	5	6
1	5	0	0	2	0	1
	0.6055	.	.	0.6251	.	0.3857
2	0	4	2	1	1	0
	.	0.6095	0.7638	0.3509	0.6607	.
3	0	1	6	1	0	0
	.	0.5520	0.7695	0.4241	.	.
4	4	0	1	3	0	0
	0.5032	.	0.7821	0.5461	.	.
5	0	3	2	0	2	1
	.	0.7723	0.5606	.	0.4897	0.6799
6	3	0	0	0	2	3
	0.6725	.	.	.	0.4296	0.5763
Total	12	8	11	7	5	5
	0.5881	0.6634	0.7316	0.5234	0.4999	0.5589
Priors	0.2000	0.2000	0.2000	0.2000	0.1000	0.1000

Zero cell counts report a missing value for the average posterior probability. We did not specify the `priors()` option with `estat classtable`, so the prior probabilities used in our LDA model were used.

`estat errorrate` estimates the error rates for each group. We use the `pp` option to obtain estimates based on the posterior probabilities instead of the counts.

```
. estat errorrate, pp
Error rate estimated from posterior probabilities
```

Error Rate	rootstock 1	2	3	4	5
Stratified	.2022195	.431596	.0868444	.4899799	.627472
Unstratified	.2404022	.41446	.1889412	.5749832	.4953118
Priors	.2	.2	.2	.2	.1

Error Rate	rootstock 6	Total
Stratified	.6416429	.3690394
Unstratified	.4027382	.3735623
Priors	.1	

We did not specify the `priors()` option, and `estat errorrate` defaulted to using the prior probabilities from the LDA model. Both stratified and unstratified estimates are shown for each rootstock group and for the overall total. See [MV] **discrim estat** for an explanation of the error-rate estimation.

We can list the classification results and posterior probabilities from our discriminant analysis model by using the `estat list` command. `estat list` allows us to specify which observations we wish to examine and what classification and probability results to report.

We request the LOO classification and LOO probabilities for all misclassified observations from the fourth rootstock group. We also suppress the resubstitution classification and probabilities from being displayed.

```
. estat list if rootstock==4, misclassified class(loo noclass) pr(loo nopr)
```

	Classification	LOO Probabilities					
Obs.	True LOO Cl.	1	2	3	4	5	6
25	4 1 *	0.5433	0.1279	0.0997	0.0258	0.0636	0.1397
26	4 3 *	0.0216	0.0199	0.7821	0.1458	0.0259	0.0048
27	4 1 *	0.3506	0.1860	0.0583	0.2342	0.0702	0.1008
29	4 1 *	0.6134	0.0001	0.0005	0.2655	0.0002	0.1202
32	4 1 *	0.5054	0.0011	0.0017	0.4856	0.0002	0.0059

`* indicates misclassified observations`

Four of the five misclassifications for rootstock group 4 were incorrectly classified as belonging to rootstock group 1.

◁

ANOVA, MANOVA, and canonical correlations

There is a mathematical relationship between Fisher's LDA and one-way MANOVA. They are both based on the eigenvalues and eigenvectors of the same matrix, $\mathbf{W}^{-1}\mathbf{B}$ (though in MANOVA the matrices are labeled $\mathbf{E}$ and $\mathbf{H}$ for error and hypothesis instead of $\mathbf{W}$ and $\mathbf{B}$ for within and between). See [MV] **manova** and [R] **anova** for more information on MANOVA and ANOVA. Researchers often wish to examine the MANOVA and univariate ANOVA results corresponding to their LDA model.

Canonical correlations are also mathematically related to Fisher's LDA. The canonical correlations between the discriminating variables and indicator variables constructed from the group variable are based on the same eigenvalues and eigenvectors as MANOVA and Fisher's LDA. The information from a canonical correlation analysis gives insight into the importance of each discriminant function in the discrimination. See [MV] **canon** for more information on canonical correlations.

The `estat manova`, `estat anova`, and `estat canontest` commands display MANOVA, ANOVA, and canonical correlation information after `discrim lda`.

▷ Example 2: MANOVA, ANOVA, and canonical correlation corresponding to LDA

Continuing with the apple tree rootstock example, we examine the MANOVA, ANOVA, and canonical correlation results corresponding to our LDA.

```
. estat manova
                                Number of obs =      48
                W = Wilks' lambda      L = Lawley-Hotelling trace
                P = Pillai's trace     R = Roy's largest root
```

Source	Statistic	df	F(df1,	df2) =	F	Prob>F	
rootstock	W 0.1540	5	20.0	130.3	4.94	0.0000	a
	P 1.3055		20.0	168.0	4.07	0.0000	a
	L 2.9214		20.0	150.0	5.48	0.0000	a
	R 1.8757		5.0	42.0	15.76	0.0000	u
Residual		42					
Total		47					

```
                e = exact, a = approximate, u = upper bound on F
. estat anova
Univariate ANOVA summaries
```

Variable	Model MS	Resid MS	Total MS	R-sq	Adj. R-sq	F	Pr > F
y1	.07356042	.31998754	.29377189	0.1869	0.0901	1.931	0.1094
y2	4.1996621	12.14279	11.297777	0.2570	0.1685	2.9052	0.0243
y3	6.1139358	4.2908128	4.484762	0.5876	0.5385	11.969	0.0000
y4	2.4930912	1.7225248	1.8044999	0.5914	0.5428	12.158	0.0000

```
                Number of obs = 48      Model df = 5      Residual df = 42
```

All four of the MANOVA tests reject the null hypothesis that the six rootstock groups have equal means. See example 1 of [MV] **manova** for an explanation of the MANOVA table.

`estat anova` presents a concise summary of univariate ANOVAs run on each of our four discriminating variables. Variables `y3`, trunk girth at 15 years, and `y4`, weight of tree above ground at 15 years, show the highest level of significance of the four variables.

`estat canontest` displays the canonical correlations and associated tests that correspond to our LDA model.

```
. estat canontest
Canonical linear discriminant analysis
```

Fcn	Canon. Corr.	Eigen-value	Variance Prop.	Variance Cumul.	Like-lihood Ratio	F	df1	df2	Prob>F	
1	0.8076	1.87567	0.6421	0.6421	0.1540	4.9369	20	130.3	0.0000	a
2	0.6645	.790694	0.2707	0.9127	0.4429	3.1879	12	106.1	0.0006	a
3	0.4317	.229049	0.0784	0.9911	0.7931	1.6799	6	82	0.1363	e
4	0.1591	.025954	0.0089	1.0000	0.9747	.54503	2	42	0.5839	e

```
Ho: this and smaller canon. corr. are zero;  e = exact F, a = approximate F
```

The number of nonzero eigenvalues in Fisher's LDA is $\min(g-1, p)$ With $g = 6$ groups, and $p = 4$ discriminating variables, there are four nonzero eigenvalues. The four eigenvalues and the corresponding canonical correlations of $\mathbf{W}^{-1}\mathbf{B}$, ordered from largest to smallest, are reported along with the proportion and cumulative proportion of variance accounted for by each of the discriminant functions. Using one discriminant dimension is insufficient for capturing the variability of our four-dimensional data. With two dimensions we account for 91% of the variance. Using three of the four dimensions accounts for 99% of the variance. Little is gained from the fourth discriminant dimension.

Also presented are the likelihood-ratio tests of the null hypothesis that each canonical correlation and all smaller canonical correlations from this model are zero. The letter a is placed beside the p-values of the approximate F tests, and the letter e is placed beside the p-values of the exact F tests. The first two tests are highly significant, indicating that the first two canonical correlations are likely not zero. The third test has a p-value of 0.1363, so that we fail to reject that the third and fourth canonical correlation are zero.

◁

Discriminant and classification functions

See [MV] **discrim lda** for a discussion of linear discriminant functions and linear classification functions for LDA.

Discriminant functions are produced from Fisher's LDA. The discriminant functions provide a set of transformations from the original p-dimensional (the number of discriminating variables) space to the minimum of p and $g - 1$ (the number of groups minus 1) dimensional space. The discriminant functions are ordered in importance.

Classification functions are by-products of the Mahalanobis approach to LDA. There are always g classification functions—one for each group. They are not ordered by importance, and you cannot use a subset of them for classification.

A table showing the discriminant function coefficients is available with `estat loadings` (see example 3), and a table showing the classification function coefficients is available with `estat classfunctions` (see example 4).

▷ Example 3: Canonical discriminant functions and canonical structures

We continue with the apple tree rootstock example. The canonical discriminant function coefficients (loadings) are available through the `estat loadings` command. Unstandardized, pooled within-group standardized, and total-sample standardized coefficients are available. The `all` option requests all three, and the `format()` option provides control over the numeric display format used in the tables.

```
. estat loadings, all format(%6.2f)
Canonical discriminant function coefficients
```

	func~1	func~2	func~3	func~4
y1	-3.05	-1.14	-1.00	23.42
y2	1.70	-1.22	1.67	-3.08
y3	-4.23	7.17	3.05	-2.01
y4	0.48	-11.52	-5.51	3.10
_cons	15.45	-12.20	-9.99	-12.47

```
Standardized canonical discriminant function coefficients
```

	func~1	func~2	func~3	func~4
y1	-0.27	-0.10	-0.09	2.04
y2	0.92	-0.65	0.90	-1.65
y3	-1.35	2.29	0.97	-0.64
y4	0.10	-2.33	-1.12	0.63

```
Total-sample standardized canonical discriminant function coefficients
```

	func~1	func~2	func~3	func~4
y1	-0.28	-0.10	-0.09	2.14
y2	1.00	-0.72	0.99	-1.81
y3	-1.99	3.37	1.43	-0.95
y4	0.14	-3.45	-1.65	0.93

The unstandardized canonical discriminant function coefficients shown in the first table are the function coefficients that apply to the unstandardized discriminating variables—y1 through y4 and a constant term. See example 5 for a graph, known as a score plot, that plots the observations transformed by these unstandardized canonical discriminant function coefficients.

The standardized canonical discriminant function coefficients are the coefficients that apply to the discriminating variables after they have been standardized by the pooled within-group covariance. These coefficients are appropriate for interpreting the importance and relationship of the discriminating variables within the discriminant functions. See example 5 for a graph, known as a loading plot, that plots these standardized coefficients.

The total-sample standardized canonical discriminant function coefficients are the coefficients that apply to the discriminating variables after they have been standardized by the total-sample covariance. See *Methods and formulas* of [MV] **discrim lda** for references discussing which of within-group and total-sample standardization is most appropriate.

For both styles of standardization, variable y1 has small (in absolute value) coefficients for the first three discriminant functions. This indicates that y1 does not play an important part in these discriminant functions. Because the fourth discriminant function accounts for such a small percentage of the variance, we ignore the coefficients from the fourth function when assessing the importance of the variables.

Some sources, see Huberty (1994), advocate the interpretation of structure coefficients, which measure the correlation between the discriminating variables and the discriminant functions, instead of standardized discriminant function coefficients; see the discussion in example 1 of [MV] **discrim lda** for references to this dispute. The `estat structure` command displays structure coefficients.

```
. estat structure, format(%9.6f)
Canonical structure
```

	function1	function2	function3	function4
y1	-0.089595	-0.261416	0.820783	0.499949
y2	-0.086765	-0.431180	0.898063	0.006158
y3	-0.836986	-0.281362	0.457902	-0.103031
y4	-0.793621	-0.572890	0.162901	-0.124206

Using structure coefficients for interpretation, we conclude that y1 is important for the second and third discriminant functions.

◁

▷ Example 4: LDA classification functions

Switching from Fisher's approach to LDA to Mahalanobis's approach to LDA, we examine what are called classification functions with the `estat classfunctions` command. Classification functions are applied to the unstandardized discriminating variables. The classification function that results in the largest value for an observation indicates the group to assign the observation.

Continuing with the rootstock LDA, we specify the `format()` option to control the display format of the classification coefficients.

```
. estat classfunctions, format(%8.3f)
Classification functions
```

	rootstock					
	1	2	3	4	5	6
y1	314.640	317.120	324.589	307.260	316.767	311.301
y2	-59.417	-63.981	-65.152	-59.373	-65.826	-63.060
y3	149.610	168.161	154.910	147.652	168.221	160.622
y4	-161.178	-172.644	-150.356	-153.387	-172.851	-175.477
_cons	-301.590	-354.769	-330.103	-293.427	-349.847	-318.099
Priors	0.200	0.200	0.200	0.200	0.100	0.100

The prior probabilities, used in constructing the coefficient for the constant term, are displayed as the last row in the table. We did not specify the `priors()` option, so the prior probabilities defaulted to those in our LDA model, which has rootstock group 5 and 6 with prior probabilities of 0.1, whereas the other groups have prior probabilities of 0.2.

See example 10 for applying the classification function to data by using the `predict` command. ◁

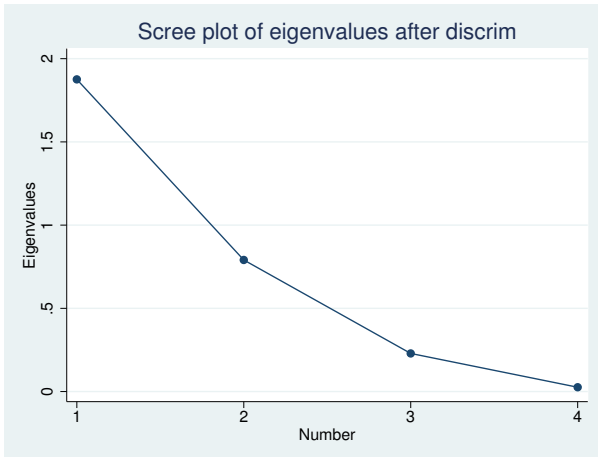
Scree, loading, and score plots

Examples of discriminant function loading plots and score plots (see [MV] **scoreplot**) can be found in example 3 of [MV] **discrim lda** and example 1 of [MV] **candisc**. Also available after `discrim lda` are scree plots; see [MV] **screeplot**.

▷ Example 5: Scree, loading, and score plots

Continuing with our rootstock example, the scree plot of the four nonzero eigenvalues we previously saw in the output of `estat canontest` in example 2 are graphed using the `screeplot` command.

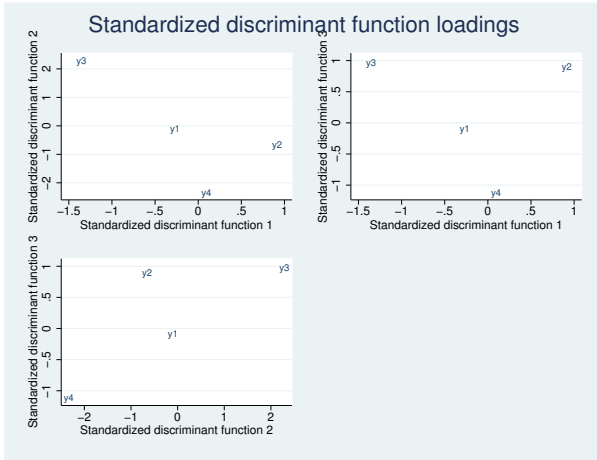
```
. screeplot
```



The *Remarks and examples* in [MV] **screeplot** concerning the use of scree plots for selecting the number of components in the context of `pca` apply also for selecting the number of discriminant functions after `discrim lda`. With these four eigenvalues, it is not obvious whether to choose the top two or three eigenvalues. From the `estat canontest` output of example 2, the first two discriminant functions account for 91% of the variance, and three discriminant functions account for 99% of the variance.

The `loadingplot` command (see [MV] **scoreplot**) allows us to graph the pooled within-group standardized discriminant coefficients (loadings) that we saw in tabular form from the `estat loadings` command of example 3. By default only the loadings from the first two functions are graphed. We override this setting with the `components(3)` option, obtaining graphs of the first versus second, first versus third, and second versus third function loadings. The `combined` option switches from a matrix graph to a combined graph. The `msymbol(i)` option removes the plotting points, leaving the discriminating variable names in the graph, and the option `mlabpos(0)` places the discriminating variable names in the positions of the plotted points.

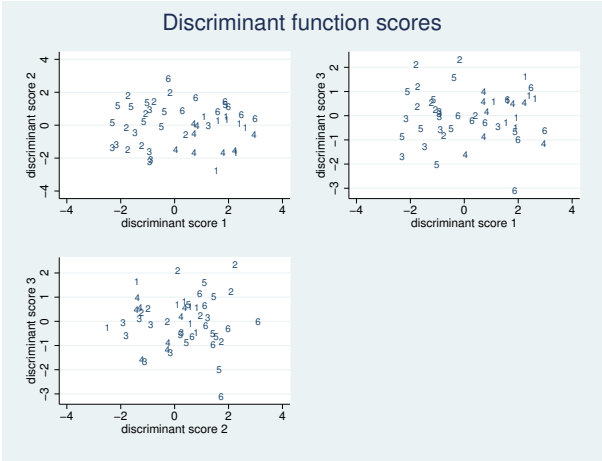
```
. loadingplot, components(3) combined msymbol(i) mlabpos(0)
```



Variable `y1`, trunk girth at 4 years, is near the origin in all three graphs, indicating that it does not play a strong role in discriminating among our six rootstock groups. `y4`, weight of tree above ground at 15 years, does not play much of a role in the first discriminant function but does in the second and third discriminant functions.

The corresponding three score plots are easily produced with the `scoreplot` command; see [MV] **scoreplot**. Score plots graph the discriminant function–transformed observations (called scores).

```
. scoreplot, components(3) combined msymbol(i)
```



There is a lot of overlap, but some separation of the rootstock groups is apparent. One of the observations from group 6 seems to be sitting by itself in the bottom of the two graphs that have discriminant function 3 as the y axis. In example 11, we identify this point by using the `predict` command.

◁

Means and distances

The `estat grsummarize` command is available after all `discrim` commands and will display means, medians, minimums, maximums, standard deviations, group sizes, and more for the groups; see [MV] **discrim estat**. After `discrim lda`, the `estat grmeans` command will also display group means. It, however, has options for displaying the within-group standardized group means, the total-sample standardized group means, and the canonical discriminant functions evaluated at the group means.

▷ Example 6: Standardized group means and canonical discriminant functions at the means

We introduce the `estat grmeans` command with the iris data originally from Anderson (1935), introduced in example 3 of [MV] **discrim lda**.

```
. use http://www.stata-press.com/data/r13/iris
(Iris data)
. discrim lda seplen sepwid petlen petwid, group(iris) notable
```

The `notable` option of `discrim` suppressed the classification table.

By default, `estat grmeans` displays a table of the means of the discriminating variables for each group. You could obtain the same information along with other statistics with the `estat grsummarize` command; see [MV] **discrim estat**.

```
. estat grmeans
Group means
```

	iris		
	setosa	versico~r	virginica
seplen	5.006	5.936	6.588
sepwid	3.428	2.77	2.974
petlen	1.462	4.26	5.552
petwid	.246	1.326	2.026

Differences in the iris species can be seen within these means. For instance, the petal lengths and widths of the *Iris setosa* are smaller than those of the other two species. See example 1 of [MV] **discrim estat** for further exploration of these differences.

The main purpose of `estat grmeans` is to present standardized or transformed means. The `totalstd` and `withinstd` options request the two available standardizations.

```
. estat grmeans, totalstd withinstd
Total-sample standardized group means
```

	iris		
	setosa	versico~r	virginica
seplen	-1.011191	.1119073	.8992841
sepwid	.8504137	-.6592236	-.1911901
petlen	-1.30063	.2843712	1.016259
petwid	-1.250704	.1661774	1.084526

```
Pooled within-group standardized group means
```

	iris		
	setosa	versico~r	virginica
seplen	-1.626555	.1800089	1.446546
sepwid	1.091198	-.8458749	-.2453234
petlen	-5.335385	1.166534	4.16885
petwid	-4.658359	.6189428	4.039416

The first table presents the total-sample standardized group means on the discriminating variables. These are the means for each group on the total-sample standardized discriminating variables.

The second table presents the pooled within-group standardized means on the discriminating variables. Instead of using the total-sample variance, the pooled within-group variance is used to standardize the variables. Of most interest in the context of an LDA is the within-group standardization.

The `canonical` option of `estat grmeans` displays the discriminant functions evaluated at the group means and gives insight into what the functions do to the groups.

```
. estat grmeans, canonical
Group means on canonical variables
```

iris	function1	function2
setosa	-7.6076	-.215133
versicolor	1.825049	.7278996
virginica	5.78255	-.5127666

The first function places *Iris setosa* strongly negative and *Iris virginica* strongly positive with *Iris versicolor* in between. The second function places *Iris virginica* and *Iris setosa* negative and *Iris versicolor* positive.

◁

The Mahalanobis distance between the groups in an LDA helps in assessing which groups are similar and which are different.

▷ Example 7: Mahalanobis distance between groups

Continuing with the iris example, we use the `estat grdistances` command to view the squared Mahalanobis distances between the three iris species.

```
. estat grdistances
Mahalanobis squared distances between groups
```

iris	iris		
	setosa	versicolor	virginica
setosa	0		
versicolor	89.864186	0	
virginica	179.38471	17.201066	0

Iris setosa is farthest from *Iris virginica* with a squared Mahalanobis distance of 179. *Iris versicolor* and *Iris virginica* are closest with a squared Mahalanobis distance of 17.

Are these distances significant? Requesting F statistics and p-values associated with these Mahalanobis squared distances between means will help answer that question. The `mahalanobis()` option requests F tests, p-values, or both.

```
. estat grdistances, mahalanobis(f p)
Mahalanobis squared distances between groups
```

Key
Mahalanobis squared distance F with 4 and 144 df p-value

iris	iris		
	setosa	versicolor	virginica
setosa	0		
	0		
	1		
versicolor	89.864186	0	
	550.18889	0	
	3.902e-86	1	
virginica	179.38471	17.201066	0
	1098.2738	105.31265	0
	9.20e-107	9.515e-42	1

All three of the means are statistically significantly different from one another.

The generalized squared distance between groups starts with the Mahalanobis squared distance between groups and adjusts for prior probabilities when they are not equal. With equal prior probabilities there will be no difference between the generalized squared distance and Mahalanobis squared distance. The `priors()` option specifies the prior probabilities for calculating the generalized squared distances.

To illustrate, we select prior probabilities of 0.2 for *I. setosa*, 0.3 for *I. versicolor*, and 0.5 for *I. virginica*.

```
. estat grdistances, generalized priors(.2, .3, .5)
Generalized squared distances between groups
                  | iris
iris              |     setosa  versicolor   virginica
------------------+---------------------------------------
           setosa |  3.2188758   92.272131   180.77101
       versicolor |  93.083061   2.4079456   18.587361
        virginica |  182.60359   19.609012   1.3862944
```

This matrix is not symmetric and does not have zeros on the diagonal.

◁

Covariance and correlation matrices

Equal group covariance matrices is an important assumption underlying LDA. The `estat covariance` command displays the group covariance matrices, the pooled within-group covariance matrix, the between-groups covariance matrix, and the total-sample covariance matrix. The `estat correlation` command provides the corresponding correlation matrices, with an option to present p-values with the correlations.

▷ Example 8: Group covariances and correlations

Continuing our examination of LDA on the iris data, we request to see the pooled within-group covariance matrix and the covariance matrices for the three iris species.

```
. estat covariance, within groups
Pooled within-group covariance matrix
                  |    seplen     sepwid     petlen     petwid
------------------+--------------------------------------------
           seplen |   .2650082
           sepwid |   .0927211   .1153878
           petlen |   .1675143   .0552435   .1851878
           petwid |   .0384014   .0327102   .0426653   .0418816
Group covariance matrices
  iris: setosa
                  |    seplen     sepwid     petlen     petwid
------------------+--------------------------------------------
           seplen |    .124249
           sepwid |   .0992163   .1436898
           petlen |   .0163551    .011698   .0301592
           petwid |   .0103306    .009298   .0060694   .0111061
  iris: versicolor
                  |    seplen     sepwid     petlen     petwid
------------------+--------------------------------------------
           seplen |   .2664327
           sepwid |   .0851837   .0984694
           petlen |    .182898   .0826531   .2208163
           petwid |   .0557796   .0412041    .073102   .0391061
  iris: virginica
                  |    seplen     sepwid     petlen     petwid
------------------+--------------------------------------------
           seplen |   .4043429
           sepwid |   .0937633   .1040041
           petlen |   .3032898   .0713796   .3045878
           petwid |   .0490939   .0476286   .0488245   .0754327
```

All variables have positive covariance—not surprising for physical measurements (length and width).

We could have requested the between-groups covariance matrix and the total-sample covariance matrix. Options of `estat covariance` control how the covariance matrices are displayed.

Correlation matrices are also easily displayed. With `estat correlations` we show the pooled within-group correlation matrix, and add the `p` option to request display of p-values with the correlations. The p-values help us evaluate whether the correlations are statistically significant.

```
. estat corr, p
Pooled within-group correlation matrix
```

Key
Correlation Two-sided p-value

	seplen	sepwid	petlen	petwid
seplen	1.00000			
sepwid	0.53024 0.00000	1.00000		
petlen	0.75616 0.00000	0.37792 0.00000	1.00000	
petwid	0.36451 0.00001	0.47053 0.00000	0.48446 0.00000	1.00000

All correlations are statistically significant. The largest correlation is between the petal length and the sepal length.

◁

Predictions

The `predict` command after `discrim lda` has options for obtaining classifications, probabilities, Mahalanobis squared distances from observations to group means, and the leave-one-out (LOO) estimates of all of these. You can also obtain the discriminant scores and classification scores for observations. The predictions can be obtained in or out of sample.

▷ Example 9: Out-of-sample LDA classification and probabilities

We use the riding-mower data from Johnson and Wichern (2007) introduced in example 1 of [MV] **discrim** to illustrate out-of-sample prediction of classification and probabilities after an LDA.

```
. use http://www.stata-press.com/data/r13/lawnmower2
(Johnson and Wichern (2007) Table 11.1)
. discrim lda lotsize income, group(owner) notable
```

Now we see how the LDA model classifies observations with income of \$90,000, \$110,000, and \$130,000, each with a lot size of 20,000 square feet. We add 3 observations to the bottom of our dataset containing these values and then use `predict` to obtain the classifications and probabilities.

```
. input
        owner  income  lots~e
 25. .  90 20
 26. . 110 20
 27. . 130 20
 28. end
. predict grp in 25/L, class
(24 missing values generated)
. predict pr* in 25/L, pr
(24 missing values generated)
. list in 25/L
```

	owner	income	lotsize	grp	pr1	pr2
25.	.	90.0	20.0	0	.5053121	.4946879
26.	.	110.0	20.0	1	.1209615	.8790385
27.	.	130.0	20.0	1	.0182001	.9818

The observation with income of $90,000 was classified as a nonowner, but it was a close decision with probabilities of 0.505 for nonowner and 0.495 for owner. The two other observations, with $110,000 and $130,000 income, were classified as owners, with higher probability of ownership for the higher income.

◁

The `estat list`, `estat classtable`, and `estat errorrate` commands (see [MV] **discrim estat**) obtain their information by calling `predict`. The LOO listings and tables from these commands are obtained by calling `predict` with the `looclass` and `loopr` options.

In addition to predictions and probabilities, we can obtain the classification scores for observations.

▷ Example 10: Classification scores

In example 4, we used the `estat classfunctions` command to view the classification functions for the LDA of the apple tree rootstock data. We can use `predict` to obtain the corresponding classification scores—the classification function applied to observations.

```
. use http://www.stata-press.com/data/r13/rootstock, clear
(Table 6.2 Rootstock Data, Rencher and Christensen (2012))
. discrim lda y1 y2 y3 y4, group(rootstock) priors(.2,.2,.2,.2,.1,.1) notable
. predict clscr*, clscore
. format clscr* %6.1f
. list rootstock clscr* in 1/3, noobs
```

rootst~k	clscr1	clscr2	clscr3	clscr4	clscr5	clscr6
1	308.1	303.7	303.1	307.1	303.5	307.1
1	327.6	324.1	322.9	326.1	323.3	326.0
1	309.5	308.2	306.3	309.3	307.5	309.0

We did not specify the `priors()` option, so `predict` used the prior probabilities that were specified with our LDA model in determining the constant term in the classification function; see example 4 for a table of the classification functions. Observations may be classified to the group with largest score. The first 3 observations belong to rootstock group 1 and are successfully classified as belonging to group 1 because the classification score in `clscr1` is larger than the classification scores for the other groups.

◁

Scoring the discriminating variables by using Fisher's canonical discriminant functions is accomplished with the `dscore` option of `predict`.

▷ Example 11: Scoring the discriminant variables

Using the rootstock data in example 5, we noticed 1 observation, from group 6, near the bottom of the score plot where the third discriminant function was the y axis. The observation has a score for the third discriminant function that appears to be below -3. We will use the `dscore` option of `predict` to find the observation.

```
. predict ds*, dscore
. format ds* %5.0g
. list rootstock y* ds* if ds3 < -3
```

	rootst~k	y1	y2	y3	y4	ds1	ds2	ds3	ds4
42.	6	0.75	0.840	3.14	0.606	1.59	1.44	-3.11	-1.93

Observation 42 is the one producing that third discriminant score.

◁

Stored results

`estat anova` stores the following in `r()`:

Scalars

`r(N)`	number of observations
`r(df_m)`	model degrees of freedom
`r(df_r)`	residual degrees of freedom

Matrices

`r(anova_stats)`	ANOVA statistics for the model

`estat canontest` stores the following in `r()`:

Scalars

`r(N)`	number of observations
`r(N_groups)`	number of groups
`r(k)`	number of variables
`r(f)`	number of canonical discriminant functions

Matrices

`r(stat)`	canonical discriminant statistics

`estat classfunction` stores the following in `r()`:

Matrices

`r(C)`	classification function matrix
`r(priors)`	group prior probabilities

`estat correlations` stores the following in `r()`:

Matrices

`r(Rho)`	pooled within-group correlation matrix (`within` only)
`r(P)`	two-sided p-values for pooled within-group correlations (`within` and `p` only)
`r(Rho_between)`	between-groups correlation matrix (`between` only)
`r(P_between)`	two-sided p-values for between-groups correlations (`between` and `p` only)
`r(Rho_total)`	total-sample correlation matrix (`total` only)
`r(P_total)`	two-sided p-values for total-sample correlations (`total` and `p` only)
`r(Rho_#)`	group # correlation matrix (`groups` only)
`r(P_#)`	two-sided p-values for group # correlations (`groups` and `p` only)

`estat covariance` stores the following in `r()`:

Matrices

`r(S)`	pooled within-group covariance matrix (`within` only)
`r(S_between)`	between-groups covariance matrix (`between` only)
`r(S_total)`	total-sample covariance matrix (`total` only)
`r(S_#)`	group # covariance matrix (`groups` only)

`estat grdistances` stores the following in `r()`:

Scalars

`r(df1)`	numerator degrees of freedom (`mahalanobis` only)
`r(df2)`	denominator degrees of freedom (`mahalanobis` only)

Matrices

`r(sqdist)`	Mahalanobis squared distances between group means (`mahalanobis` only)
`r(F_sqdist)`	F statistics for tests that the Mahalanobis squared distances between group means are zero (`mahalanobis` only)
`r(P_sqdist)`	p-value for tests that the Mahalanobis squared distances between group means are zero (`mahalanobis` only)
`r(gsqdist)`	generalized squared distances between group means (`generalized` only)

`estat grmeans` stores the following in `r()`:

Matrices

`r(means)`	group means (`raw` only)
`r(stdmeans)`	total-sample standardized group means (`totalstd` only)
`r(wstdmeans)`	pooled within-group standardized group means (`withinstd` only)
`r(cmeans)`	group means on canonical variables (`canonical` only)

`estat loadings` stores the following in `r()`:

Matrices

`r(L_std)`	Within-group standardized canonical discriminant function coefficients (`standardized` only)
`r(L_totalstd)`	total-sample standardized canonical discriminant function coefficients (`totalstandardized` only)
`r(L_unstd)`	unstandardized canonical discriminant function coefficients (`unstandardized` only)

`estat manova` stores the following in `r()`:

Scalars

`r(N)`	number of observations
`r(df_m)`	model degrees of freedom
`r(df_r)`	residual degrees of freedom

Matrices

`r(stat_m)`	multivariate statistics for the model

`estat structure` stores the following in `r()`:

Matrices
`r(canstruct)` canonical structure matrix

Methods and formulas

See *Methods and formulas* of [MV] **discrim lda** for background on what is produced by `predict`, `estat classfunctions`, `estat grdistances`, `estat grmeans`, `estat loadings`, and `estat structure`. See [MV] **discrim estat** for more information on `estat classtable`, `estat errorrate`, `estat grsummarize`, and `estat list`. See [R] **anova** for background information on the ANOVAs summarized by `estat anova`; see [MV] **manova** for information on the MANOVA shown by `estat manova`; and see [MV] **canon** for background information on canonical correlations and related tests shown by `estat canontest`.

References

Anderson, E. 1935. The irises of the Gaspe Peninsula. *Bulletin of the American Iris Society* 59: 2–5.

Andrews, D. F., and A. M. Herzberg, ed. 1985. *Data: A Collection of Problems from Many Fields for the Student and Research Worker*. New York: Springer.

Huberty, C. J. 1994. *Applied Discriminant Analysis*. New York: Wiley.

Johnson, R. A., and D. W. Wichern. 2007. *Applied Multivariate Statistical Analysis*. 6th ed. Englewood Cliffs, NJ: Prentice Hall.

Rencher, A. C., and W. F. Christensen. 2012. *Methods of Multivariate Analysis*. 3rd ed. Hoboken, NJ: Wiley.

Also see

[MV] **discrim lda** — Linear discriminant analysis

[MV] **discrim estat** — Postestimation tools for discrim

[MV] **scoreplot** — Score and loading plots

[MV] **screeplot** — Scree plot

[MV] **candisc** — Canonical linear discriminant analysis

[MV] **canon** — Canonical correlations

[MV] **discrim** — Discriminant analysis

[MV] **manova** — Multivariate analysis of variance and covariance

[U] **20 Estimation and postestimation commands**