Package ‘DiscriMiner’

February 19, 2015

Type Package

Title Tools of the Trade for Discriminant Analysis

Version 0.1-29

Date 2013-11-14

Depends R (>= 2.15.0)

Suggests MASS, FactoMineR

Description Functions for Discriminant Analysis and Classification purposes covering various methods such as descriptive, geometric, linear, quadratic, PLS, as well as qualitative discriminant analyses

License GPL-3

URL http://www.gastonsanchez.com

LazyData yes

Collate 'DiscriMiner-package.R' 'FRatio.R' 'betweenCov.R'
'betweenSS.R' 'binarize.R' 'classify.R' 'corRatio.R' 'desDA.R'
'discPower.R' 'disqual.R' 'easyMCA.R' 'geoDA.R' 'getWithin.R'
'groupMeans.R' 'groupMedians.R' 'groupQuants.R' 'groupStds.R'
'groupVars.R' 'linDA.R' 'my_betweenCov.R' 'my_catDA.R'
'my_discFunctions.R' 'my_geoDA.R' 'my_groupMeans.R'
'my_linDA.R' 'my_mca.R' 'my_plsDA.R' 'my_quaDA.R' 'my_tdc.R'
'my_verify.R' 'my_withinCov.R' 'plot.plsda.R' 'print.desda.R'
'print.disqual.R' 'print.geoda.R' 'print.linda.R'
'print.plsda.R' 'print.quada.R' 'print.qualmca.R' 'quaDA.R'
'totalCov.R' 'totalSS.R' 'withinCov.R' 'withinSS.R'
'my_plsDA_old.R' 'plsDA_old.R' 'plsDA.R'

Author Gaston Sanchez [aut, cre], Charles Determan [ctb]

Maintainer Gaston Sanchez <gaston.stat@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2013-11-15 00:52:30
Description

DiscriMiner contains several functions for Discriminant Analysis and Classification purposes covering various methods such as descriptive, geometric, linear, quadratic, PLS, as well as qualitative discriminant analyses.
betweenCov

Details

Package: DiscriMiner
Type: Package
Version: 0.1-23
Date: 2012-12-20
License: GPL-3

Author(s)

Gaston Sanchez
Maintainer: Gaston Sanchez <gaston.stat@gmail.com>

References

http://www.gastonsanchez.com/discriminer

betweenCov

Between-class Covariance Matrix

Description

Calculates between-class covariance matrix

Usage

betweenCov(variables, group, div_by_n = FALSE)
### betweenSS

**Between-class Sum of Squares Matrix**

**Description**

Calculates between-class sum of squares and cross product matrix (a.k.a. between-class scatter matrix)

**Usage**

```r
betweenSS(variables, group)
```

**Arguments**

- **variables**: matrix or data frame with explanatory variables (No missing values are allowed)
- **group**: vector or factor with group membership (No missing values are allowed)

**Details**

When `div_by_n` is `TRUE`, the covariance matrices are divided by `n` (number of observations), otherwise they are divided by `n-1`

**Author(s)**

Gaston Sanchez

**See Also**

`getWithin`, `betweenSS`, `withinCov`, `totalCov`

**Examples**

```r
## Not run:
# load iris dataset
data(iris)

# between-class covariance matrix (dividing by n-1)
betweenCov(iris[,1:4], iris[,5])

# between-class covariance matrix (dividing by n)
betweenCov(iris[,1:4], iris[,5], div_by_n=TRUE)

## End(Not run)
```
binarize

Author(s)
Gaston Sanchez

See Also
betweenCov, withinSS, totalSS

Examples
## Not run:
# load iris dataset
data(iris)

# between-class scatter matrix
betweenSS(iris[,1:4], iris[,5])

## End(Not run)

binarize Binarize a data frame into a super-indicator matrix

Description
Convert a data frame with factors into a super-indicator matrix (a.k.a. complete disjunctive table from the french tableau disjonctive complete)

Usage
binarize(variables)

Arguments
variables data frame with categorical variables (coded as factors)

Value
A super-indicator matrix of binary data

Author(s)
Gaston Sanchez

References
See Also
easyMCA

Examples

```r
## Not run:
# load insurance cars dataset
data(insurance)

# super-indicator matrix of binary data
bin_insure = binarize(insurance[, -1])
head(bin_insure)

## End(Not run)
```

---

**bordeaux**

*Bordeaux Wines Dataset*

Description

Quality measures of wines from Bordeaux, France

Format

A data frame with 34 observations on the following 6 variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>year of harvest</td>
</tr>
<tr>
<td>temperature</td>
<td>sum of daily average temperatures (in celsius degrees)</td>
</tr>
<tr>
<td>sun</td>
<td>duration of insolation (in hours)</td>
</tr>
<tr>
<td>heat</td>
<td>number of super-hot days</td>
</tr>
<tr>
<td>rain</td>
<td>rain level (in millimeters)</td>
</tr>
<tr>
<td>quality</td>
<td>wine quality: a factor with levels bad, good, and medium</td>
</tr>
</tbody>
</table>

References

Chapter 10: Analyse Discriminante, page 353.

Examples

```r
## Not run:
# load data
data(bordeaux)

# structure of data
str(bordeaux)

## End(Not run)
```
classify  

Classification function  

Description  
Classify provided observations based on a given Discriminant object  

Usage  
classify(DA_object, newdata)  

Arguments  
DA_object  
discriminant analysis object  
newdata  
vector or matrix or data frame with variables for which their classes will be calculated  

Details  
A DA_object is a discriminant analysis (DA) object obtained from a geometric predictive DA (class "geoda"), a linear DA (class "linda"), a quadratic DA (class "quada"), or a DISQUAL analysis (class "disqual")  

Value  
A list with the following elements  
scores  
discriminant scores for each observation  
pred_class  
predicted class  

Author(s)  
Gaston Sanchez  

See Also  
geoDA, linDA, quaDA, plsDA, disqual  

Examples  
## Not run:  
# load iris dataset  
data(iris)  

# linear discriminant analysis  
my_lin1 = linDA(iris[,1:4], iris$Species)  

# select a sample of 15 observations
```r
set.seed(111)
obs = sample(1:nrow(iris), 15)
some_data = iris[obs, 1:4]

# classify some_data
get_classes = classify(my_lin1, some_data)
get_classes

# compare the results against original class
table(iris$Species[obs], get_classes$pred_class)

## End(Not run)
```

---

**corRatio**

*Correlation Ratio*

**Description**

Calculates the correlation ratio between a quantitative variable and a qualitative variable.

**Usage**

```r
corRatio(variable, group)
```

**Arguments**

- `variable`: a single quantitative variable
- `group`: vector or factor with group memberships (qualitative variable)

**Details**

No missing values are allowed.

**Author(s)**

Gaston Sanchez

**References**


**See Also**

`FRatio`, `discPower`
Examples

```r
## Not run:
# iris dataset
data(iris)

# correlation ratio between Petal-Length and Species
corRatio(iris$Petal.Length, iris$Species)
```

## End(Not run)

---

**Descriptive Discriminant Analysis**

**Description**

Performs a Descriptive Discriminant Analysis (a.k.a. Factorial Discriminant Analysis from the french Analyse Factorielle Discriminante)

**Usage**

```r
desDA(variables, group, covar = "within")
```

**Arguments**

- `variables`: matrix or data frame with explanatory variables
- `group`: vector or factor with group memberships
- `covar`: character string indicating the covariance matrix to be used. Options are "within" and "total"

**Details**

When `covar="within"` the estimated pooled within-class covariance matrix is used in the calculations.
When `covar="total"` the total covariance matrix is used in the calculations.
The difference between `covar="within"` and `covar="total"` is in the obtained eigenvalues.
The estimated pooled within-class covariance matrix is actually the within-class covariance matrix divided by the number of observations minus the number of classes (see `getWithin`)

**Value**

An object of class "desda", basically a list with the following elements

- `power`: table with discriminant power of the explanatory variables
- `values`: table of eigenvalues
- `discrivar`: table of discriminant variables, i.e. the coefficients of the linear discriminant functions
- `discor`: table of correlations between the variables and the discriminant axes
- `scores`: table of discriminant scores for each observation
**discPower**

**Description**

Measures Discriminant Power of explanatory variables

**Usage**

```r
discPower(variables, group)
```
discPower

Arguments

variables  matrix or data frame with explanatory variables

group  vector or factor with group membership

Details

No missing values are allowed

Value

A data frame containing the following columns

correl_ratio  Correlation Ratios
wilks_lambda  Wilks Lambda
F_statistic  F-statistic
p_value  p-value of F-statistic

Author(s)

Gaston Sanchez

References


See Also

corRatio, FRatio

Examples

```r
## Not run:
# bordeaux wines dataset
data(bordeaux)

# discriminant power
dp = discPower(bordeaux[,2:5], bordeaux$quality)
dp

## End(Not run)
```
**DISQUAL**

**Description**

Implementation of the DISQUAL methodology. Disqual performs a Fishers Discriminant Analysis on components from a Multiple Correspondence Analysis.

**Usage**

```r
disqual(variables, group, validation = NULL,
         learn = NULL, test = NULL, autosel = TRUE, prob = 0.05)
```

**Arguments**

- `variables`: data frame with qualitative explanatory variables (coded as factors)
- `group`: vector or factor with group memberships
- `validation`: type of validation, either "crossval" or "learntest". Default NULL
- `learn`: optional vector of indices for a learn-set. Only used when `validation="learntest"`. Default NULL
- `test`: optional vector of indices for a test-set. Only used when `validation="learntest"`. Default NULL
- `autosel`: logical indicating automatic selection of MCA components
- `prob`: probability level for automatic selection of MCA components. Default `prob = 0.05`

**Details**

- When `validation=NULL` there is no validation
- When `validation="crossval"` cross-validation is performed by randomly separating the observations in ten groups.
- When `validation="learntest"` validation is performed by providing a learn-set and a test-set of observations.

**Value**

An object of class "disqual", basically a list with the following elements:

- `raw_coefs`: raw coefficients of discriminant functions
- `norm_coefs`: normalized coefficients of discriminant functions, ranging from 0 - 1000
- `confusion`: confusion matrix
- `scores`: discriminant scores for each observation
- `classification`: assigned class
- `error_rate`: misclassification error rate
easyMCA

Description

Performs a basic Multiple Correspondence Analysis (MCA)

Usage

easyMCA(variables)

Arguments

variables data frame with categorical variables (coded as factors)
FRatio

Description
Calcualtes the F-statistic between a quantitative variable and a qualitative variable.

Usage
FRatio(variable, group)

Arguments
variable a quantitative variable
group a vector or factor with group memberships (i.e. qualitative variable)
**geoDA**

**Value**

F-statistic and its p-value

**Author(s)**

Gaston Sanchez

**References**


**See Also**

`discPower, corRatio`

**Examples**

```r
## Not run:
# load bordeaux wines dataset
data(bordeaux)

# F-statistic ratio between temperature and quality
FRatio(bordeaux$temperature, bordeaux$quality)

## End(Not run)
```

---

**geoDA**

*Geometric Predictive Discriminant Analysis*

**Description**

Performs a Geometric Predictive Discriminant Analysis

**Usage**

```r
geoDA(variables, group, validation = NULL, learn = NULL, test = NULL)
```

**Arguments**

- `variables`: matrix or data frame with explanatory variables
- `group`: vector or factor with group memberships
- `validation`: type of validation, either "crossval" or "learntest". Default NULL
- `learn`: optional vector of indices for a learn-set. Only used when `validation="learntest"`. Default NULL
- `test`: optional vector of indices for a test-set. Only used when `validation="learntest"`. Default NULL
Details

When validation=NULL there is no validation
When validation="crossval" cross-validation is performed by randomly separating the observations in ten groups.
When validation="learntest" validation is performed by providing a learn-set and a test-set of observations.

Value

An object of class "geoda", basically a list with the following elements:

- functions: table with discriminant functions
- confusion: confusion matrix
- scores: discriminant scores for each observation
- classification: assigned class
- error_rate: misclassification error rate

Author(s)

Gaston Sanchez

References


See Also

classify, desDA, linDA, quaDA, plsDA

Examples

```r
## Not run:
# load bordeaux wines dataset
data(iris)

# geometric predictive discriminant analysis with no validation
my_geo1 = geoDA(iris[,1:4], iris$Species)
my_geo1$confusion
my_geo1$error_rate

# geometric predictive discriminant analysis with cross-validation
my_geo2 = geoDA(iris[,1:4], iris$Species, validation="crossval")
my_geo2$confusion
my_geo2$error_rate
```
getWithin

## Description
 Calculates the estimated within-class covariance matrix

## Usage
 `getWithin(variables, group)`

## Arguments
- `variables`: matrix or data frame with explanatory variables (No missing values are allowed)
- `group`: vector or factor with group memberships (No missing values are allowed)

## Details
 The obtained matrix is the estimated within-class covariance matrix (i.e. within-class covariance matrix divided by its degrees of freedom $n-k$, where $n$ is the number of observations and $k$ is the number of groups)

## Author(s)
 Gaston Sanchez

## See Also
 `withinCov`

## Examples
```r
## Not run:
# load iris dataset
data(iris)

# estimated within-class covariance matrix (dividing by n-k)
getWithin(iris[,1:4], iris[,5])

# compared to the within-class covariance matrix (dividing by n-1)
withinCov(iris[,1:4], iris[,5])

## Not run
```
**Description**

Calculates means for each group

**Usage**

```
groupMeans(variables, group, na.rm = FALSE)
```

**Arguments**

- `variables` matrix or data frame with explanatory variables (may contain missing values)
- `group` vector or factor with group memberships
- `na.rm` logical indicating whether missing values should be removed

**Value**

matrix of group means (with variables in the rows, and groups in the columns)

**Author(s)**

Gaston Sanchez

**See Also**

`groupVars`, `groupStdS`, `groupMedians`, `groupQuants`

**Examples**

```r
## Not run:
# dataset iris
data(iris)

# group means
groupMeans(iris[,1:4], iris[,5])

## End(Not run)
```
### groupMedians

**Group Medians**

#### Description

Calculates the medians for each group

#### Usage

```
groupMedians(variables, group, na.rm = FALSE)
```

#### Arguments

- **variables**: matrix or data frame with explanatory variables (may contain missing values)
- **group**: vector or factor with group memberships
- **na.rm**: logical indicating whether missing values should be removed

#### Value

matrix of group medians (with variables in the rows, and groups in the columns)

#### Author(s)

Gaston Sanchez

#### See Also

- `groupVars`, `groupStd`, `groupMeans`, `groupQuants`

#### Examples

```r
## Not run:
# dataset iris
data(iris)

# group means
groupMedians(iris[,1:4], iris[,5])

## End(Not run)
```
**groupQuants**

**Group Quantiles**

**Description**

Calculates the specified quantiles for each group

**Usage**

```r
groupQuants(variables, group, prob, na.rm = FALSE)
```

**Arguments**

- `variables`: matrix or data frame with explanatory variables (may contain missing values)
- `group`: vector or factor with group memberships
- `prob`: probability value (numeric value between 0 and 1)
- `na.rm`: logical indicating whether missing values should be removed

**Value**

matrix of group quantiles (with variables in the rows, and groups in the columns)

**Author(s)**

Gaston Sanchez

**See Also**

`groupMeans`, `groupVars`, `groupStats`, `groupMedians`

**Examples**

```r
## Not run:
# dataset iris
data(iris)

# group quantile prob=20
groupQuants(iris[,1:4], iris[,5], prob=0.20)

## End(Not run)
```
**groupStds**

| Description |  
|-------------|---
| Calculates the standard deviations for each group |

**Usage**

```r
groupStds(variables, group, na.rm = FALSE)
```

**Arguments**

- `variables`: matrix or data frame with explanatory variables (may contain missing values)
- `group`: vector or factor with group memberships
- `na.rm`: logical indicating whether missing values should be removed

**Value**

matrix of group standard deviations (with variables in the rows, and groups in the columns)

**Author(s)**

Gaston Sanchez

**See Also**

`groupMeans`, `groupVars`, `groupMedians`, `groupQuants`

**Examples**

```r
## Not run:
# dataset iris
data(iris)

# group standard deviations
groupStds(iris[,1:4], iris[,5])

## End(Not run)
```
groupVars  

*Group Variances*

**Description**

Calculates the variances for each group

**Usage**

```
groupVars(variables, group, na.rm = FALSE)
```

**Arguments**

- `variables`  
  matrix or data frame with explanatory variables (may contain missing values)
- `group`  
  vector or factor with group memberships
- `na.rm`  
  logical indicating whether missing values should be removed

**Value**

matrix of group variances (with variables in the rows, and groups in the columns)

**Author(s)**

Gaston Sanchez

**See Also**

`groupMeans`, `groupStd`, `groupMedians`, `groupQuants`

**Examples**

```
## Not run:
# dataset iris
data(iris)

# group variances
groupVars(iris[,1:4], iris[,5])

## End(Not run)
```
**infarctus**  
*Infarctus dataset*

**Description**

Infarctus dataset from Saporta (2006)

**Format**

A data frame with 101 observations on the following 8 variables.

- **FRCAR** Frequence Cardiaque (i.e. heart rate)
- **INCAR** Index Cardique (cardiac index)
- **INSYS** Index Systolique (systolic index)
- **PRDIA** Pression Diastolique (diastolic pressure)
- **PAPUL** Pression Arterielle Pulmonaire (pulmonary artery pressure)
- **PVENT** Pression Ventriculaire (ventricular pressure)
- **REPUL** Resistance Pulmonaire (pulmonary resistance)
- **PRONO** Pronostic (prognosis): a factor with levels dead and survive

**References**

Chapter 18: Analyse discriminante et regression logistique, pp 453-454  

**Examples**

```r
## Not run:
# load data
data(infarctus)

# summary of variables
summary(infarctus)

## End(Not run)
```

**insurance**  
*Insurance Dataset*

**Description**

Dataset of car-insurance customers from Belgium in 1992

**Format**

A data frame with 1106 observations on the following 10 variables.
Claims Group variable. A factor with levels bad and good
Use Type of Use. A factor with levels private and professional
Type Insurance Type. A factor with levels companies, female, and male
Language Language. A factor with levels flemish and french
BirthCohort Birth Cohort. A factor with levels BD_1890_1949, BD_1950_1973, and BD_unknown
Region Geographic Region. A factor with levels Brussels and Other_regions
BonusMalus Level of bonus-malus. A factor with levels BM_minus and BM_plus
YearSuscrip Year of Subscription. A factor with levels YS<86 and YS>=86
Horsepower Horsepower. A factor with levels HP<=39 and HP>=40
YearConstruc Year of vehicle construction. A factor with levels YC_33_89 and YC_90_91

Details

Dataset for DISQUAL method

References


See Also
disqual

Examples

```r
## Not run:
# load data
data(insurance)

# structure
str(insurance)

## End(Not run)
```

linDA Linear Discriminant Analysis

Description

Performs a Linear Discriminant Analysis

Usage

```r
linDA(variables, group, prior = NULL, validation = NULL,
      learn = NULL, test = NULL, prob = FALSE)
```
linDA

Arguments

variables: matrix or data frame with explanatory variables

group: vector or factor with group memberships

prior: optional vector of prior probabilities. Default prior=NULL implies group proportions

validation: type of validation, either "crossval" or "learntest". Default NULL

learn: optional vector of indices for a learn-set. Only used when validation="learntest". Default NULL

test: optional vector of indices for a test-set. Only used when validation="learntest". Default NULL

prob: logical indicating whether the group classification results should be expressed in probability terms

Details

When validation=NULL there is no validation
When validation="crossval" cross-validation is performed by randomly separating the observations in ten groups.
When validation="learntest" validation is performed by providing a learn-set and a test-set of observations.

Value

An object of class "linda", basically a list with the following elements:

functions: table with discriminant functions

confusion: confusion matrix

scores: discriminant scores for each observation

classification: assigned class

error_rate: misclassification error rate

Author(s)

Gaston Sanchez

References


See Also

classify, desDA, geoDA, quaDA, plsDA
Examples

```r
## Not run:
# load iris dataset
data(iris)

# linear discriminant analysis with no validation
my_lin1 = linDA(iris[,1:4], iris$Species)
my_lin1$confusion
my_lin1$error_rate

# linear discriminant analysis with cross-validation
my_lin2 = linDA(iris[,1:4], iris$Species, validation="crossval")
my_lin2$confusion
my_lin2$error_rate

# linear discriminant analysis with learn-test validation
learning = c(1:40, 51:90, 101:140)
testing = c(41:50, 91:100, 141:150)
my_lin3 = linDA(iris[,1:4], iris$Species, validation="learntest", learn=learning, test=testing)
my_lin3$confusion
my_lin3$error_rate

## End(Not run)
```

---

**plsDA**

*PLS Discriminant Analysis*

**Description**

Performs a Partial Least Squares (PLS) Discriminant Analysis by giving the option to include a random leave-k fold out cross validation

**Usage**

```r
plsDA(variables, group, autosel = TRUE, comps = 2,
validation = NULL, learn = NULL, test = NULL,
cv = "LOO", k = NULL, retain.models = FALSE)
```

**Arguments**

- **variables**: matrix or data frame with explanatory variables
- **group**: vector or factor with group memberships
- **autosel**: logical indicating automatic selection of PLS components by cross-validation. Default autosel=TRUE
- **comps**: integer greater than one indicating the number of PLS components to retain. Used only when autosel=FALSE
- **validation**: type of validation, either NULL or "learntest". Default NULL
learn optional vector of indices for a learn-set. Only used when validation="learntest". Default NULL

test optional vector of indices for a test-set. Only used when validation="learntest". Default NULL

cv string indicating the type of crossvalidation. Available options are "L00" (Leave-One-Out) and "LKO" (Leave-K fold-Out)
k fold left out if using LKO (usually 7 or 10)
retain.models whether to retain lower models (i.e. all lower component results)

Details
When validation=NULL leave-one-out (loo) cross-validation is performed.
When validation="learntest" validation is performed by providing a learn-set and a test-set of observations.

Value
An object of class "plsd"a", basically a list with the following elements:

functions table with discriminant functions
confusion confusion matrix
scores discriminant scores for each observation
loadings loadings
y.loadings y loadings
classification assigned class
error_rate misclassification error rate
components PLS components
Q2 quality of loo cross-validation
R2 R-squared coefficients
VIP Variable Importance for Projection
comp-vars correlations between components and variables
comp_group correlations between components and groups

Author(s)
Charles Determan Jr, Gaston Sanchez

References
See Also
classify, geoDA, linDA, quaDA

Examples

```r
## Not run:
# load iris dataset
data(iris)

# PLS discriminant analysis specifying number of components = 2
my_pls1 = plsDA(iris[,1:4], iris$Species, autosel=FALSE, comps=2)
my_pls1$confusion
my_pls1$error_rate
# plot circle of correlations
plot(my_pls1)

# PLS discriminant analysis with automatic selection of components
my_pls2 = plsDA(iris[,1:4], iris$Species, autosel=TRUE)
my_pls2$confusion
my_pls2$error_rate

# linear discriminant analysis with learn-test validation
learning = c(1:40, 51:90, 101:140)
testing = c(41:50, 91:100, 141:150)
my_pls3 = plsDA(iris[,1:4], iris$Species, validation="learntest",
learn=learning, test=testing)
my_pls3$confusion
my_pls3$error_rate

## End(Not run)
```

quada

**Quadratic Discriminant Analysis**

Description

Performs a Quadratic Discriminant Analysis

Usage

```r
quada(variables, group, prior = NULL, validation = NULL,
       learn = NULL, test = NULL, prob = FALSE)
```

Arguments

- `variables`: matrix or data frame with explanatory variables
- `group`: vector or factor with group memberships
- `prior`: optional vector of prior probabilities. Default `prior=NULL` implies group proportions
validation: type of validation, either "crossval" or "learntest". Default NULL
learn: optional vector of indices for a learn-set. Only used when validation="learntest". Default NULL
test: optional vector of indices for a test-set. Only used when validation="learntest". Default NULL
prob: logical indicating whether the group classification results should be expressed in probability terms

Details
When validation=NULL there is no validation
When validation="crossval" cross-validation is performed by randomly separating the observations in ten groups.
When validation="learntest" validation is performed by providing a learn-set and a test-set of observations.

Value
An object of class "quada", basically a list with the following elements:

- confusion: confusion matrix
- scores: discriminant scores for each observation
- classification: assigned class
- error_rate: misclassification error rate

Author(s)
Gaston Sanchez

References

See Also
classify, desDA, geoDA, linDA, plsDA

Examples
```r
## Not run:
# load iris dataset
data(iris)

# quadratic discriminant analysis with no validation
```
my_qua1 = quaDA(iris[,1:4], iris$Species)
my_qua1$confusion
my_qua1$error_rate

# quadratic discriminant analysis with cross-validation
my_qua2 = quaDA(iris[,1:4], iris$Species, validation="crossval")
my_qua2$confusion
my_qua2$error_rate

# quadratic discriminant analysis with learn-test validation
learning = c(1:40, 51:90, 101:140)
testing = c(41:50, 91:100, 141:150)
my_qua3 = quaDA(iris[,1:4], iris$Species, validation="learntest",
  learn=learning, test=testing)
my_qua3$confusion
my_qua3$error_rate

## End(Not run)

---

**totalCov**  
*Total Covariance Matrix*

---

**Description**

Calculates total covariance matrix

**Usage**

```r
totalCov(variables, div_by_n = FALSE)
```

**Arguments**

- `variables`: matrix or data frame with explanatory variables (No missing values are allowed)
- `div_by_n`: logical indicating division by number of observations

**Details**

When `div_by_n=TRUE` the covariance matrices are divided by n (number of observations), otherwise they are divided by n-1

**Author(s)**

Gaston Sanchez

**See Also**

`totalSS, betweenCov, withinCov`
**Examples**

```r
## Not run:
# load iris dataset
data(iris)

# total covariance matrix (dividing by n-1)
totalCov(iris[,1:4])

# total covariance matrix (dividing by n)
totalCov(iris[,1:4], div_by_n=TRUE)

## End(Not run)
```

---

**totalSS**

*Total Sum of Squares Matrix*

---

**Description**

Calculates the total sum of squares and cross product matrix (a.k.a. total scatter matrix)

**Usage**

`totalSS(variables)`

**Arguments**

- `variables` matrix or data frame with explanatory variables

**Author(s)**

Gaston Sanchez

**See Also**

`totalCov, betweenSS, withinSS`

**Examples**

```r
## Not run:
# load iris dataset
data(iris)

# total scatter matrix
totalSS(iris[,1:4])

## End(Not run)
```
withinCov  

**Within-class Covariance Matrix**

**Description**
Calculates the within-class covariance matrix

**Usage**

```r
withinCov(variables, group, div_by_n = FALSE)
```

**Arguments**

- `variables`: matrix or data frame with explanatory variables (No missing values are allowed)
- `group`: vector or factor with group memberships (No missing values are allowed)
- `div_by_n`: logical indicating division by number of observations

**Details**
When `div_by_n` = TRUE the covariance matrices are divided by n (number of observations), otherwise they are divided by n-1

**Author(s)**
Gaston Sanchez

**See Also**

`withinSS, betweenCov, totalCov`

**Examples**

```r
## Not run:
# load iris dataset
data(iris)

# within-class covariance matrix (dividing by n-1)
withinCov(iris[,1:4], iris[,5])

# within-class covariance matrix (dividing by n)
withinCov(iris[,1:4], iris[,5], div_by_n=TRUE)

## End(Not run)
```
withinSS

Within-class Sum of Squares Matrix

Description
Calculates within-class sum of squares and cross product matrix (a.k.a. within-class scatter matrix)

Usage
withinSS(variables, group)

Arguments
variables matrix or data frame with explanatory variables (No missing values are allowed)
group vector or factor with group memberships (No missing values are allowed)

Author(s)
Gaston Sanchez

See Also
withinCov, betweenSS, totalSS

Examples
## Not run:
# load iris dataset
data(iris)

# within-class scatter matrix
withinSS(iris[,1:4], iris[,5])

## End(Not run)
Index

*Topic datasets
  bordeaux, 6
  infarctus, 23
  insurance, 23

*Topic package
  DiscriMiner-package, 2
  betweenCov, 3, 5, 30, 32
  betweenSS, 4, 4, 31, 33
  binarize, 5, 13, 14
  bordeaux, 6
  classify, 7, 13, 16, 25, 28, 29
  corRatio, 8, 11, 15
  desDA, 9, 16, 25, 29
  discPower, 8, 10, 10, 15
  DiscriMiner-package, 2
  disqual, 7, 12, 14, 24
  easyMCA, 6, 13, 13
  FRatio, 8, 11, 14
  geoDA, 7, 15, 25, 28, 29
  getWithin, 4, 9, 17
  groupMeans, 18, 19–22
  groupMedians, 18, 19, 20–22
  groupQuants, 18, 19, 20, 21, 22
  groupStds, 18–20, 21, 22
  groupVars, 18–21, 22
  infarctus, 23
  insurance, 23
  linDA, 7, 16, 24, 28, 29
  plsDA, 7, 16, 25, 26, 29
  quaDA, 7, 16, 25, 28, 28
  totalCov, 4, 30, 31, 32
  totalSS, 5, 30, 31, 33
  withinCov, 4, 17, 30, 32, 33
  withinSS, 5, 31, 32, 33