

Package ‘Dprep’

April 7, 2008

Type Package

Title Data preprocessing and visualization functions for classification

Version 2.0

Date 2008-04-07

Author Edgar Acuna<edgar@cs.uprm.edu>, and members of the CASTLE group at UPR-Mayaguez.

Maintainer Edgar Acuna <edgar@math.uprm.edu>

Description Functions for normalization, handling of missing values, discretization, outlier detection, feature selection, and visualization

Depends MASS, nnet, lattice, class

Suggests rpart

License GPL

R topics documented:

dprep-package	3
assign	3
baysout	4
breastw	5
bupa	6
ce.impute	6
ce.knn.imp	7
ce.mimp	8
censusn	9
chiMerge	10
circledraw	11
clean	12
closest	13
colon	14
combinations	14
crossval	15
cv10knn2	16
cv10lda2	16
cv10log	17

cv10mlp	18
cv10rpart2	19
decscale	19
diabetes	20
disc.1r	21
disc.ef	22
disc.ew	23
disc.mentr	23
disc2	24
discretevar	25
dist.to.knn	25
distan2	26
distancia	26
ec.knnimp	27
eje1dis	28
finco	28
hawkins	29
heartc	30
hepatitis	31
imagmiss	32
inconsist	33
ionosphere	34
knneigh.vect	35
lofactor	35
lvf	36
mahaout	37
mardia	38
maxdist	39
maxlof	39
midpoints	40
mmnorm	41
mo3	42
mo4	42
moda	43
my.iris	44
near1	45
near2	45
nnmiss	46
outbox	46
parallelplot	47
pp.golub	48
radviz2d	49
rangenorm	50
reachability	51
redundancy	52
relief	53
reliefcat	54
reliefcont	54
robot	55
row.matches	56
sbs1	56
score	57

<i>assig</i>	3
sffs	57
sfs	58
sfs1	59
signorm	60
softmaxnorm	61
sonar	61
srbct	62
starcoord	63
surveyplot	64
tchisq	65
top	65
vehicle	66
vvalen	67
vvalen1	68
znorm	68
Index	70

dprep-package	<i>Data Preprocessing for Supervised Classification</i>
---------------	---

Description

Functions for normalization, treatment of missing values, discretization, outlier detection, feature selection, and visualization

Details

Package: dprep
 Type: Package
 Version: 2.0
 Date: 2008-04-07
 License: GPL (version 2 or later) See file LICENCE.

For an overview of how to use the package, see user manual provided.

Author(s)

Maintainer: Edgar Acuna <edgar@cs.uprm.edu>

<i>assig</i>	<i>Auxiliary function for computing the minimum entropy discretization</i>
--------------	--

Description

This function is used in the computation of the minimum entropy discretization

Usage

```
assig(x, points, nparti, n)
```

Arguments

<code>x</code>	A given vector
<code>points</code>	The number of points
<code>nparti</code>	The number of partitions
<code>n</code>	The number of points

Author(s)

Luis Daza

See Also

[disc.mentr](#)

baysout

Outlier detection using Bay and Schwabacher's algorithm.

Description

This function implements the algorithm for outlier detection found in Bay and Schwabacher(2003). The algorithm assigns an outlyingness measure to each observation and returns the indexes of those observations having the largest measures. The number of outliers to be returned is specified by the user.

Usage

```
baysout(D, blocks = 5, k = 3, num.out = 10)
```

Arguments

<code>D</code>	the dataset under study
<code>blocks</code>	the number of sections in which to divide the entire dataset. It must be at least as large as the number of outliers requested.
<code>k</code>	the number of neighbors to find for each observation
<code>num.out</code>	the number of outliers to return

Value

`num.out` Returns a two column matrix containing the indexes of the observations with the top `num.out` outlyingness measures. A plot of the top candidates and their measures is also displayed.

Author(s)

Caroline Rodriguez(2004). Modified by Elio Lozano (2005)

References

Bay, S.D., and Schwabacher (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule.

Examples

```
#---- Outliers detection using the Bay's algorithm----
data(bupa)
bupa.out=baysout(bupa[bupa[,7]==1,1:6],blocks=10,num.out=10)
```

breastw

*The Breast Wisconsin dataset***Description**

This is the Breast Wisconsin dataset from the UCI Machine Learning Repository. Sixteen instances with missing values have been deleted from the original dataset.

Usage

```
data(breastw)
```

Format

A data frame with 683 observations on the following 10 variables.

- V1** Clump Thickness
- V2** Uniformity of Cell Size
- V3** Uniformity of Cell Shape
- V4** Marginal Adhesion
- V5** Single Epithelial Cell Size
- V6** Bare Nuclei
- V7** Bland Chromatin
- V8** Normal Nucleoli
- V9** Mitoses
- V10** Class: 1 for benign, 2 for Malign

Details

All the features assume values in the range 1-10. The original dataset contains 699 observations but 16 of them have been delete because contain missing values

Source

The UCI Machine Learning Database Repository at:

- <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>
- <http://www.ics.uci.edu/~mlearn/MLRepository.html>

Examples

```
#Detecting outliers in class-1 using the LOF algorithms---
data(breastw)
maxlof(breastw[breastw[,10]==1,],name="",30,40)
```

bupa

The Bupa dataset

Description

The Bupa dataset

Usage

```
data(bupa)
```

Format

A data frame with 345 observations on the following 7 variables.

- V1** mean corpuscular volume
- V2** alkaline phosphotase
- V3** alamine aminotransferase
- V4** aspartate aminotransferase
- V5** gamma-glutamyl transpeptidase
- V6** number of half-pint equivalents of alcoholic beverages drunk per day
- V7** The class variable (two classes)

Source

The UCI Machine Learning Database Repository at:

- <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>
- <http://www.ics.uci.edu/~mllearn/MLRepository.html>

Examples

```
#---Sequential forward feature selection using the lda classifier---  
data(bupa)  
sfs(bupa, "lda", repet=10)
```

ce.impute*Imputation in supervised classification*

Description

This function performs data imputation in datasets for supervised classification by using mean, median or knn imputation methods.

Usage

```
ce.impute(data, method = c("mean", "median", "knn"), atr,  
nomatr = rep(0, 0), k1 = 10)
```

Arguments

data	the name of the dataset
method	the name of the method to be used
atr	a vector identifying the attributes where imputations will be performed
nomatr	a vector identifying the nominal attributes
k1	the number of neighbors to be used for the knn imputation

Value

Returns a matrix without missing values.

Note

A description of all the imputations carried out may be stored in a report that is later saved to the current workspace. To produce the report, lines at the end of the code must be uncommented. The report objects name starts with Imput.rep.

Author(s)

Caroline Rodriguez

References

Acuna, E. and Rodriguez, C. (2004). The treatment of missing values and its effect in the classifier accuracy. In D. Banks, L. House, F.R. McMorris, P. Arabie, W. Gaul (Eds). Classification, Clustering and Data Mining Applications. Springer-Verlag Berlin-Heidelberg, 639-648.

See Also

[clean](#)

Examples

```
data(hepatitis)
#-----Median Imputation-----
#ce.impute(hepatitis, "median", 1:19)
#-----knn Imputation-----
hepa.imputed=ce.impute(hepatitis, "knn", k1=10)
```

ce.knn.imp

Function that calls ec.knnimp to perform knn imputation

Description

This function simply sets up the dataset so that missing values can be imputed by knn imputation. ec.knnimp is the function that actually carries out the imputation.

Usage

```
ce.knn.imp(m, natr = rep(0, 0), k1)
```

Arguments

<code>m</code>	matrix containing relevant variables and classes
<code>natr</code>	list of nominal attributes
<code>k1</code>	number of neighbors to use for imputation

Value

<code>r</code>	matrix with missing values imputed
----------------	------------------------------------

Author(s)

Caroline Rodriguez and Edgar Acuna

Examples

```
data(hepatitis)
hepa.knnimp=ce.knn.imp(hepatitis,natr=c(1,3:14),k1=10)
```

ce.mimp

Mean or median imputation

Description

A function that detects the location of missing values by class, then imputes the missing values that occur in the features, using mean or median imputation, as selected by the user. If the feature is nominal then imputation is done using the mode.

Usage

```
ce.mimp(w.cl, method = c("mean", "median"), atr, nomatr = 0)
```

Arguments

<code>w.cl</code>	dataset with missing values.
<code>method</code>	either "mean" or "median"
<code>atr</code>	list of relevant features
<code>nomatr</code>	list of nominal features, imputation is done using mode

Value

<code>w.cl</code>	the original matrix with values imputed
-------------------	---

Note

A description of all the imputations carried out may be stored in a report that is later saved to the current workspace. To produce the report, lines at the end of the code must be uncommented. The report objects name starts with `Imput.rep`.

Author(s)

Caroline Rodriguez and Edgar Acuna

References

Acuna, E. and Rodriguez, C. (2004). The treatment of missing values and its effect in the classifier accuracy. In D. Banks, L. House, F.R. McMorris, P. Arabie, W. Gaul (Eds). Classification, Clustering and Data Mining Applications. Springer-Verlag Berlin-Heidelberg, 639-648.

Examples

```
data(hepatitis)
#-----Mean Imputation-----
hepa.mean.imp=ce.impute(hepatitis, "mean", 1:19)
```

censusn

The census dataset

Description

This is the census dataset from the UCI where the values of the nominal attributes are numerically codified. This dataset contains plenty of missing values.

Usage

```
data(censusn)
```

Format

A data frame with 32561 observations on the following 14 variables.

- V1** age:continuous
- V2** workclass:
- V3** fnlwgt:continuous
- V4** education
- V5** marital-status:
- V6** occupation:
- V7** relationship:
- V8** race
- V9** sex
- V10** capital-gain: continuous.
- V11** capital-loss: continuous.
- V12** hours-per-week: continuous.
- V13** native-country:
- V14** class: >50K, <=50K

Details

The fifth and fourth features of the original dataset were the same, since the fifth contained the numerical codifications of the fourth. In censusn only one of these feature is considered. The values of the nominal attributes are as follows: workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked. education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse. occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces. relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried. race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black. sex: Female, Male. native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinidad&Tobago, Peru, Hong, Holand-Netherlands.

Source

The UCI Machine Learning Database Repository at:

- <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>
- <http://www.ics.uci.edu/~mlearn/MLRepository.html>

Examples

```
data(censusn)
#----knn imputation-----
data(censusn)
imagmiss(censusn, "censusn")
```

chiMerge

Discretization using the Chi-Merge method

Description

This function performs supervised discretization using the Chi Merge method.

Usage

```
chiMerge(data, varcon, alpha = 0.1)
```

Arguments

data	The name of the dataset to be discretized
varcon	Vector of continuous variables
alpha	The significance level

Details

In case of datasets containing negative values apply first a range normalization to change the range of the attributes values to an interval containing positive values. The discretization process becomes slow when the number of variables increases (say for more than 100 variables).

Value

`discdata` A new data matrix containing the discretized features

Author(s)

Edgar Acuna, Jaime Porras, and Carlos Lopez

References

Kantardzic M. (2003). Data Mining: Concepts, Models, methods, and Algorithms. John Wiley. New York.

See Also

[disc.ef](#), [disc.ew](#), [disc.lr](#), [disc.mentr](#)

Examples

```
#-----Discretization using the ChiMerge method
data(my.iris)
iris.disc=chiMerge(my.iris,1:4,alpha=0.05)
#-----Applying chiMerge a dataset containing negative values
#data(ionosphere)
#normionos=rangenorm(ionosphere,"mmnorm")
#ionos.disc=chiMerge(normionos,1:32)
```

`circledraw`

circledraw

Description

This function draws a circle using the polygon function from the graphics package. It is an auxiliary function used by `radviz2d`.

Usage

```
circledraw (numpts = 200, radius = 1)
```

Arguments

`numpts` Number of edges of the polygon, default is 200.
`radius` Radius of the circle to be drawn, default is 1.

Details

A circle of a specified radius is drawn by the polygon function of the graphics library by constructing a polygon with numpts number of edges. It is intended to be an auxiliary function for the radviz2d visualization.

Value

Displays a circle of radius = radius.

Author(s)

Caroline Rodriguez

Examples

```
#----Circledraw examples
circledraw()
```

clean

Dataset Cleaning

Description

A function to eliminate rows and columns that have a percentage of missing values greater than the allowed tolerance.

Usage

```
clean(w, tol.col = 0.5, tol.row = 0.3, name = "")
```

Arguments

w	the dataset to be examined and cleaned
tol.col	maximum ratio of missing values allowed in columns. The default value is 0.5. Columns with a larger ratio of missing will be eliminated unless they are known to be relevant attributes.
tol.row	maximum ratio of missing values allowed in rows. The default value is 0.3. Rows with a ratio of missing that is larger than the established tolerance will be eliminated.
name	name of the dataset to be used for the optional report

Details

This function can create an optional report on the cleaning process if the comment symbols are removed from the last lines of code. The report is returned to the workspace, where it can be reexamined as needed. The report object's name begins with: Clean.rep.

Value

w the original dataset, with missing values that were in relevant variables imputed

Author(s)

Caroline Rodriguez

References

Acuna, E. and Rodriguez, C. (2004). The treatment of missing values and its effect in the classifier accuracy. In D. Banks, L. House, F.R. McMorris, P. Arabie, W. Gaul (Eds). Classification, Clustering and Data Mining Applications. Springer-Verlag Berlin-Heidelberg, 639-648.

See Also

[ce.impute](#)

Examples

```
#-----Dataset cleaning-----  
data(hepatitis)  
hepa.cl=clean(hepatitis,0.5,0.3,name="hepatitis-clean")
```

closest

Auxiliary function used in the function baysout

Description

Function used by baysout to select the k vectors that are closer to a given instance.

Usage

```
closest(dis, neigh, k)
```

Arguments

dis	An instance from the dataset under study
neigh	A matrix containing the distance from the given observations to each of its k neighbors.
k	The number of nearest neighbors

Author(s)

Caroline Rodriguez

See Also

[baysout](#)

 colon

Alon et al.'s colon dataset

Description

This is Alon et al.'s Colon cancer dataset which contains information on 62 samples for 2000 genes. The samples belong to tumor and normal colon tissues.

Usage

```
data(colon)
```

Format

A data frame with 62 observations for 2000 genes. An additional column contains the tissue classes.

Source

The data is available at:

- <http://microarray.princeton.edu/oncology/>

References

Alon U, Barkai N, Notterman DA, Gish, K, Ybarra, S, Mack, D and Levine, AJ. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. USA. 96. p. 6745-6750.

Examples

```
#Detecting the top 5 outliers in class-2 using the LOF algorithm
data(colon)
colon2.lof=maxlof(colon[colon[,2001]==2,], "colon-class2")
colon2.lof[order(colon2.lof,decreasing=TRUE)][1:5]
```

 combinations

Constructing distinct permutations

Description

A function for constructing the minimal set of permutations of the elements in the sequence 1:numcol as described by Wegman in Hyperdimensional Data Analysis(1999)

Usage

```
combinations(numcol)
```

Arguments

numcol A value representing the number of columns in a matrix

Value

A matrix in which each column represents a distinct permutation of the sequence 1:numcol

Author(s)

Caroline Rodriguez

References

Wegman, E. (1990), Hyperdimensional data analysis using parallel coordinates, Journal of the American Statistical Association, 85, 664-675.

crossval

Cross validation estimation of the misclassification error

Description

Computation of the misclassification error for the LDA, KNN and rpart classifiers by cross validation

Usage

```
crossval(data, nparts = 10, method = c("lda", "knn", "rpart"),  
kvec = 5, repet)
```

Arguments

data	The name of the dataset
nparts	The number of folds in which the dataset is divided. By default nparts=10.
method	The name of the classifier to be used: LDA,KNN, Rpart.
kvec	The number of nearest neighbors to be used for the KNN classifier.
repet	The number of repetitions

Value

Returns the mean misclassification crossvalidation error of the classifier obtained on a given number of repetitions

Author(s)

Edgar Acuna

See Also

[cv10log](#), [cv10mlp](#)

Examples

```
#-----10-fold crossvalidation error using the LDA classifier---
data(bupa)
crossval(bupa,method="lda",repet=10)
#-----5-fold crossvalidation error using the knn classifier---
data(colon)
crossval(colon,nparts=5,method="knn",kvec=3,repet=5)
```

cv10knn2

Auxiliary function for sequential feature selection

Description

This function finds the number of instances correctly classified by the knn classifier, using 10-fold cross validation, with one repetition

Usage

```
cv10knn2(data, kvec)
```

Arguments

data	The name of the dataset
kvec	The number of neighbors

Author(s)

Edgar Acuna

See Also

[crossval](#)

cv10lda2

Auxiliary function for sequential forward selection

Description

This function finds the number of instances correctly classified by the Linear Discriminant classifier using 10 fold cross validation with one repetition.

Usage

```
cv10lda2(data)
```

Arguments

data	The name of the dataset
------	-------------------------

Author(s)

Edgar Acuna

See Also[crossval](#)

cv10log	<i>10-fold cross validation estimation error for the classifier based on logistic regression</i>
---------	--

Description

10-fold cross validation estimation of the misclassification error for the classifier based on logistic regression

Usage

```
cv10log(data, repet, maxwts=2500)
```

Arguments

data	The name of the dataset
repet	The number of repetitions
maxwts	The maximum number of weights to be estimated. It must be an integer greater than the number of predictors of the dataset.

Value

The mean cross validation error for the classifier based on logistic regression using the number of repetitions

Author(s)

Edgar Acuna

References

Ripley, B.D. (1996). Pattern recognition and Neural networks. Cambridge University Press
 Venables, W.N., and Ripley, B.D. (2002). Modern Applied Statistics with S. Fourth edition, Springer

See Also[crossval](#), [cv10mlp](#)**Examples**

```
#-----cross validation error for the logistic classifier-----
data(bupa)
cv10log(bupa, 5)
```

cv10mlp	<i>10-fold cross validation error estimation for the multilayer perceptron classifier</i>
---------	---

Description

10-fold cross validation estimation error for the multilayer perceptron classifier.

Usage

```
cv10mlp(data, units, decay = 0, maxwts = 1000, maxit = 100,
repet)
```

Arguments

data	The name of the dataset
units	The number of units in the hidden layer
decay	The decay parameter
maxwts	The maximum number of weights to be estimated in the network
maxit	The maximum number of iterations
repet	The number of repetitions

Value

Returns the mean cross validation for the multilayer perceptron classifier.

Author(s)

Edgar Acuna

References

Ripley, B.D. (1996). Pattern recognition and Neural networks. Cambridge University Press
Venables, W.N., and Ripley, B.D. (2002). Modern Applied Statistics with S. Fourth edition, Springer

See Also

[crossval](#), [cv10log](#)

Examples

```
#-----cross validation using the MLP classifier---
data(heartc)
cv10mlp(heartc, 25, decay=0.1, maxwts=1000, maxit=100, repet=2)
```

`cv10rpart2`*Auxiliary function for sequential feature selection*

Description

This function finds the number of instances correctly classified by the decision tree classifier, `rpart`, using 10-fold cross validation and one repetition.

Usage

```
cv10rpart2(datos)
```

Arguments

`datos` The name of the dataset

Author(s)

Edgar Acuna

See Also

[crossval](#)

`decscale`*Decimal Scaling*

Description

This is a function to apply decimal scaling to a matrix or dataframe. Decimal scaling transforms the data into $[-1,1]$ by finding k such that the absolute value of the maximum value of each attribute divided by 10^k is less than or equal to 1.

Usage

```
decscale(data)
```

Arguments

`data` The dataset to be scaled

Details

Uses the scale function found in the R base package.

Value

`decdata` The original matrix that has been scaled by decimal scaling

Author(s)

Caroline Rodriguez and Edgar Acuna

Examples

```
data(sonar)
def=par(mfrow=c(2,1))
plot(sonar[,2])
dssonar=decscale(sonar)
plot(dssonar[,2])
par(def)
```

diabetes

The Pima Indian Diabetes dataset

Description

This is the Pima Indian diabetes dataset from the UCI Machine Learning Repository.

Usage

```
data(diabetes)
```

Format

A data frame with 768 observations on the following 9 variables.

- V1** Number of times pregnant
- V2** Plasma glucose concentration (glucose tolerance test)
- V3** Diastolic blood pressure (mm Hg)
- V4** Triceps skin fold thickness (mm)
- V5** 2-Hour serum insulin (μ U/ml)
- V6** Body mass index (weight in kg/(height in m)²)
- V7** Diabetes pedigree function
- V8** Age (years)
- V9** Class variable (1:tested positive for diabetes, 0: tested negative fro diabetes)

Source

The UCI Machine Learning Database Repository at:

- <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>
- <http://www.ics.uci.edu/~mlearn/MLRepository.html>

Examples

```
#---Feature selection using SFS with the LDA classifier--
data(diabetes)
sfs(diabetes,"lda",repet=4)
```

`disc.1r`*Discretization using the Holte's 1R method*

Description

This function performs supervised discretization using the Holte's 1R method

Usage

```
disc.1r(data, convar, binsize = 6)
```

Arguments

<code>data</code>	The name of the dataset to be discretized
<code>convar</code>	A vector containing the continuous features
<code>binsize</code>	The number of instances per bin

Value

Returns a new data matrix with discretized values

Author(s)

Shiyun Wen and Edgar Acuna

References

Kantardzic M. (2003). Data Mining: Concepts, Models, methods, and Algorithms. John Wiley. New York.

See Also

[disc.ew](#), [disc.ef](#), [chiMerge](#), [disc.mentr](#)

Examples

```
#----Discretization using the Holte's 1r method
data(bupa)
disc.1r(bupa, 1:6)
```

`disc.ef`*Discretization using the method of equal frequencies*

Description

Unsupervised discretization using intervals of equal frequencies

Usage

```
disc.ef(data, varcon, k)
```

Arguments

<code>data</code>	The dataset to be discretized
<code>varcon</code>	A vector containing the continuous features
<code>k</code>	The number of intervals to be used

Value

Returns a new data matrix with discretized values.

Author(s)

Edgar Acuna

References

Kantardzic M. (2003). Data Mining: Concepts, Models, methods, and Algorithms. John Wiley. New York.

See Also

[disc.lr](#), [disc.ew](#), [chiMerge](#)

Examples

```
#Discretization using the equal frequency method
data(bupa)
bupa.disc.ef=disc.ef(bupa,1:6,8)
```

`disc.ew`*Discretization using the equal width method*

Description

Unsupervised discretization using intervals of equal width. The widths are computed using Scott's formula.

Usage

```
disc.ew(data, varcon)
```

Arguments

<code>data</code>	The name of the dataset containing the attributes to be discretized
<code>varcon</code>	A vector containing the indexes of the attributes to be discretized

Value

Returns a new data matrix with discretized values.

Author(s)

Edgar Acuna

References

Venables, W.N., and Ripley, B.D. (2002). Modern Applied Statistics with S. Fourth edition, Springer

See Also

[disc.ef](#), [disc.lr](#), [chiMerge](#), [disc.mentr](#)

Examples

```
#----Discretization using the equal frequency method
data(bupa)
bupa.disc.ew=disc.ew(bupa,1:6)
```

`disc.mentr`*Discretization using the minimum entropy criterion*

Description

This function discretizes the continuous attributes of a data frame using the minimum entropy criterion with the minimum description length as stopping rule.

Usage

```
disc.mentr(data, vars)
```

Arguments

<code>data</code>	The name of the dataset to be discretized
<code>vars</code>	A vector containing the indices of the columns to be discretized and column containing the classes

Value

Returns a matrix containing only discretized features.

Author(s)

Luis Daza

References

Dougherty, J., Kohavi, R., and Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. ML-95.

See Also

[disc.lr](#), [disc.ew](#), [disc.ef](#), [chiMerge](#)

Examples

```
data(my.iris)
iris.discme=disc.mentr(my.iris,1:5)
```

`disc2`

Auxiliary function for performing discretization using equal frequency

Description

This function is called by the `disc.ef` function in the `dprep` library.

Usage

```
disc2(x, k)
```

Arguments

<code>x</code>	A numerical vector
<code>k</code>	The number of intervals

Author(s)

Edgar Acuna

See Also

[disc.ef](#)

discretevar	<i>Performs Minimum Entropy discretization for a given attribute</i>
-------------	--

Description

This function carries out ME discretization for a given attribute of a dataset. It is also called from within the function `discr.mentr`.

Usage

```
discretevar(data, var, n, p)
```

Arguments

<code>data</code>	The name of the dataset
<code>var</code>	The column where the attribute to be discretized is located
<code>n</code>	The number of rows of the dataset
<code>p</code>	The number of columns of the dataset

Author(s)

Luis Daza

See Also

[disc.mentr](#)

<code>dist.to.knn</code>	<i>Auxiliary function for the LOF algorithm.</i>
--------------------------	--

Description

This function returns an object in which columns contain the indices of the first k neighbors followed by the distances to each of these neighbors.

Usage

```
dist.to.knn(dataset, neighbors)
```

Arguments

<code>dataset</code>	The name of the dataset
<code>neighbors</code>	The number of neighbors

Author(s)

Caroline Rodriguez

See Also

[maxlof](#)

distan2

Auxiliary function used by the RELIEF function in the dprep library.

Description

Computes the distance between two instances of a dataset considering both continuous and nominal attributes.

Usage

```
distan2(x, y, vnom)
```

Arguments

x	A given instance
y	A given instance
vnom	A vector containing the indexes of nominal attributes

Author(s)

Edgar Acuna

See Also

[relief](#)

distancia

Vector-Vector Euclidean Distance Function

Description

Finds the euclidean distance between two vectors x and y, or the vector x and the matrix y

Usage

```
distancia(x, y)
```

Arguments

x	A numeric vector
y	A numeric vector or matrix

Details

Does not support missing values.

Value

distancia	The result is a numeric value representing the Euclidean distance between x and y, or a row matrix representing the Euclidean distance between x and each row of y.
-----------	---

Author(s)

Caroline Rodriguez and Edgar Acuna

Examples

```
#---- Calculating distances
x=rnorm(4)
y=matrix(rnorm(12),4,3)
distancia(x,y[,1])
distancia(x,y)
```

ec.knnimp

KNN Imputation

Description

Function to carry out KNN imputation of missing values.

Usage

```
ec.knnimp(data, nomatr, k = 10)
```

Arguments

data	Original dataset with missing values
nomatr	Vector containing the indices of nominal attributes
k	Numeric value representing the number of neighbors to use for imputation

Details

This function is called by the function `ce.knn.imp` which is part of this library, to impute values by class. If called alone the function will impute values based on information in the entire matrix and not the classes. Needs also the function: `nnmiss`.

Value

data2	contains values belonging to one class (of a larger matrix) for which missing values in relevant variables have been imputed.
-------	---

Author(s)

Edgar Acuna and Caroline Rodriguez

References

Acuna, E. and Rodriguez, C. (2004). The treatment of missing values and its effect in the classifier accuracy. In D. Banks, L. House, F.R. McMorris, P. Arabie, W. Gaul (Eds). Classification, Clustering and Data Mining Applications. Springer-Verlag Berlin-Heidelberg, 639-648.

Examples

```
#---- Performing knn imputation----
data(hepatitis)
hepa.knnimp=ec.knnimp(hepatitis,k=10)
```

`ejeldis`*Basic example for discriminant analysis*

Description

This data frame contains information about 32 students. The first two columns contain their grades obtained on the first two exams and the last column of the dataset contains the classes: P=Pass, and F=Fail

Usage

```
data(ejeldis)
```

Format

A data frame with 32 observations on the following 3 variables.

V1 Grade on the first exam

V2 Grade on the second exam

class The class vector:P=Pass, F=Fail

Source

Data obtained from Edgar Acuna:

- <http://math.uprm.edu/~edgar/datosclass.html>

Examples

```
#---- Performing 10-fold cross validation using the LDA classifier-----  
data(ejeldis)  
crossval(ejeldis,10,"lda",repet=5)
```

`finco`*FINCO Feature Selection Algorithm*

Description

This function selects features using the FINCO algorithm. The dataset must contain only discretized values.

Usage

```
finco(data, level)
```

Arguments

<code>data</code>	Name of the dataset containing the discretized values
<code>level</code>	Minimum inconsistency level

Details

The level value must be greater than the inconsistency of the whole dataset, which first must be discretized. The function `inconsist` included in this library computes inconsistencies. A small value for level yields a greater number of selected features.

Value

<code>varselec</code>	Index of selected features
<code>inconsis</code>	Inconsistency rates of the selected features

Author(s)

Edgar Acuna

References

Acuna, E , (2003) A comparison of filters and wrappers for feature selection in supervised classification. Proceedings of the Interface 2003 Computing Science and Statistics. Vol 34.

Acuna, E., Coaquira, F. and Gonzalez, M. (2003). A comparison of feature selection procedures for classifiers based on kernel density estimation. Proc. of the Int. Conf. on Computer, Communication and Control technologies, CCCT03. VolI. p. 468-472. Orlando, Florida.

See Also

[inconsist,lvf](#)

Examples

```
#---- Feature Selection with FINCO
data(my.iris)
iris.discew=disc.ew(my.iris,1:6)
inconsist(iris.discew)
finco(iris.discew,0.05)
```

hawkins

The Hawkins-Bradu-Kass dataset

Description

An artificial dataset generated by Hawkins, Bradu, and Kass used for illustrating some of the merits of robust techniques.

Usage

```
data(hawkins)
```

Format

A data frame consisting of 75 observations on the following 4 variables.

- V1** First predictor variable
- V2** Second predictor variable
- V3** Third predictor variable
- V4** The response variable

Source

The data appears on p. 94 of Rousseeuw, P, and Leroy, A. (1987). Robust Regression and outlier detection. John Wiley & Sons. New York.

References

Hawkins, D.M, Bradu, D., Kass, G.V.(1984). Location of several outliers in multiple regression data using elemental sets. Technometrics, 26. 197-208.

Examples

```
#---- Finding outliers using the LOF algorithm----
data(hawkins)
haw.lof=maxlof(hawkins[,1:3], "Hawkins")
haw.lof[order(haw.lof,decreasing=TRUE)]
```

heartc

The Heart Cleveland dataset

Description

This dataset contains information concerning heart disease diagnosis. The data was collected from the Cleveland Clinic Foundation, and it is available at the UCI machine learning Repository. Six instances containing missing values have been deleted from the original dataset.

Usage

```
data(heartc)
```

Format

A data frame with 297 observations on the following 14 variables.

- V1** age(continuous)
- V2** sex
- V3** cp, chest pain type:1,2,3,4
- V4** trestbps: resting blood pressure(continuous)
- V5** cholesterol(continuous)
- V6** fbs: fasting blood sugar>120? yes=1, no =0
- V7** restecg: resting electrocardiographic results, 0,1, 2

- V8** thalach: maximum heart rate achieved(continuous)
- V9** exang: exercise induced angina (1 = yes; 0 = no)
- V10** oldpeak = ST depression induced by exercise relative to rest (continuous)
- V11** slope: the slope of the peak exercise ST segment
- V12** ca: number of major vessels (0-3) colored by flourosopy
- V13** thal: 3 = normal; 6 = fixed defect; 7 = reversable defect
- V14** diagnosis of heart disease: 1: < 50 2: > 50

Details

Six instances containing missing values have been deleted from the original dataset. This dataset includes continuous, binomial, nominal, and ordinal features.

Source

The UCI Machine Learning Database Repository at:

- <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>
- <http://www.ics.uci.edu/~mllearn/MLRepository.html>

Examples

```
#----Detecting outliers using the Relief---
data(heartc)
relief(heartc,100,0.4,vnom=c(2,3,6,7,9,11:13))
```

hepatitis

The hepatitis dataset

Description

This is the hepatitis dataset from the UCI. The data was donated by Gail Gong.

Usage

```
data(hepatitis)
```

Format

A data frame with 155 observations on the following 20 variables. This dataset contains a large number of missing values.

- V1** Histology:no,yes
- V2** age
- V3** sex: male,female
- V4** steroid:no,yes
- V5** antivirals:no,yes
- V6** fatigue:no, yes
- V7** malaise:no, yes

- V8 anorexia:no, yes
- V9 liver big:no,yes
- V10 liver firm:no,yes
- V11 spleen palpable: no, yes
- V12 spiders:no,yes
- V13 ascites:no,yes
- V14 Varices:no,yes
- V15 Bilirubin
- V16 alk phosphate
- V17 sgot
- V18 Albumin
- V19 Protime
- V20 Class:Die, Live

Details

The original dataset has the class labels in the first column.

Source

The UCI Machine Learning Database Repository at:

- <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>
- <http://www.ics.uci.edu/~mllearn/MLRepository.html>

References

Diaconis,P. & Efron,B. (1983). Computer-Intensive Methods in Statistics. Scientific American, Volume 248.

Examples

```
#-----Report and plot of missing values -----
data(hepatitis)
imagmiss(hepatitis, "Hepatitis")
```

imagmiss

Visualization of Missing Data

Description

Function to create a graph of the observations of the dataset leaving white gaps where data is missing.

Usage

```
imagmiss(data, name = "")
```

Arguments

data The dataset containing missing values
 name The name of dataset to be used in title of plot

Details

The main idea is to use the original dataset to create a temporary dataset containing 1 if a value is found or 0 if the value is missing. The temporary data set is graphed by column, changing color for each feature and leaving a blank horizontal line if a value is missing. Assumes classes are in the last column, and removes the column containing the classes before plotting. A report that describes the percentage of missing values in the data set is provided once the visualization is complete.

Author(s)

Caroline Rodriguez and Edgar Acuna

References

Acuna, E. and Rodriguez, C. (2004). The treatment of missing values and its effect in the classifier accuracy. In D. Banks, L. House, F.R. McMorris, P. Arabie, W. Gaul (Eds). Classification, Clustering and Data Mining Applications. Springer-Verlag Berlin-Heidelberg, 639-648.

Examples

```
#---- Plotting datasets with missing values
data(censusn)
imagmiss(censusn, "censusn")
```

 inconsist

Computing the inconsistency measure

Description

This function computes the inconsistency of a discretized dataset.

Usage

```
inconsist(data)
```

Arguments

data a discretized dataset

Details

This function requires the function row.matches included in this environment package, and the function unique from the base library.

Value

incon the inconsistency measure of the dataset

Author(s)

Edgar Acuna

References

Dash M., Liu H, and Motoda, H. (1998). Consistency Based Feature Selection Pacific-Asia Conference on Knowledge Discovery and Data Mining

See Also

[finco](#), [lvf](#)

Examples

```
##---- Calculating Inconsistency ----  
data(bupa)  
bupa.discew=disc.ew(bupa,1:6)  
inconsist(bupa.discew)
```

ionosphere

The Ionosphere dataset

Description

The Ionosphere dataset from the UCI Machine Learning Repository

Usage

```
data(ionosphere)
```

Format

A data frame with 351 observations on the following 33 variables.

Details

The original dataset contains 34 predictors, but we have eliminated the two first features, because the first feature had the same value in one of the classes and the second feature assumes the value 0 in all observations.

Source

The UCI Machine Learning Database Repository at:

- <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>
- <http://www.ics.uci.edu/~mllearn/MLRepository.html>

Examples

```
#---Outlier detection in ionosphere class-1 using the Mahalanobis distance---  
data(ionosphere)  
mahaout(ionosphere,1)
```

`knnneigh.vect`*Auxiliary function for computing the LOF measure.*

Description

Function that returns the distance from a vector "x" to its k-nearest-neighbors in the matrix "data"

Usage

```
knnneigh.vect(x, data, k)
```

Arguments

x	A given instance of the data matrix
data	The name of the data matrix
k	The number of neighbors

Author(s)

Caroline Rodriguez

See Also

[maxlof](#)

`lofactor`*Local Outlier Factor*

Description

A function that finds the local outlier factor (Breunig et al.,2000) of the matrix "data" using k neighbors. The local outlier factor (LOF) is a measure of outlyingness that is calculated for each observation. The user decides whether or not an observation will be considered an outlier based on this measure. The LOF takes into consideration the density of the neighborhood around the observation to determine its outlyingness.

Usage

```
lofactor(data, k)
```

Arguments

data	The data set to be explored
k	The kth-distance to be used to calculate the LOF's.

Details

The LOFs are calculated over a range of values, and the max local outlier factor is determined over this range.

Value

lof A vector with the local outlier factor of each observation

Author(s)

Caroline Rodriguez

References

Breuning, M., Kriegel, H., Ng, R.T, and Sander. J. (2000). LOF: Identifying density-based local outliers. In Proceedings of the ACM SIGMOD International Conference on Management of Data.

Examples

```
#---- Detecting the top 10 outliers using the LOF algorithm----
data(my.iris)
iris.lof=lofactor(my.iris,10)
```

lvf *Las Vegas Filter*

Description

Las Vegas Filter uses a random generation of subsets and an inconsistency measure as the evaluation function to determine the relevance of features in the dataset.

Usage

```
lvf(data, lambda, maxiter)
```

Arguments

data	Name of the discretized dataset
lambda	Threshold for the inconsistency
maxiter	Maximum number of iterations

Details

If the dataset has continuous variables, these must first be discretized. This package includes four discretization methods. A value of lambda close to the inconsistency of the whole dataset yields a large number of selected features, a large lambda yields few selected features.

Value

bestsubset The best subset of features

Author(s)

Edgar Acuna

References

LIU, H. and SETIONO, R. (1996). A probabilistic approach to feature selection: a filter solution. Proc. of the thirteenth International Conference of Machine Learning, 319-337.

See Also

`disc.ew,inconsist,finco`

Examples

```
#---- LVF method ----  
data(my.iris)  
iris.discew=disc.ew(my.iris,1:6)  
inconsist(iris.discew)  
lvf(iris.discew,0,300)
```

mahaout

Multivariate outlier detection through the boxplot of the Mahalanobis distance

Description

This function finds multivariate outliers by constructing a boxplot of the Mahalanobis distance of all the instances.

Usage

```
mahaout(data, nclass, plot = TRUE)
```

Arguments

<code>data</code>	Name of the dataset
<code>nclass</code>	Number of the class to check for outliers
<code>plot</code>	Logical value. If <code>plot=T</code> a plot of the mahalanobis distance is drawn

Details

uses `cov.rob` function from the MASS library

Value

Returns a list of top outliers according to their Mahalanobis distance and a list of all the instances ordered according to their Mahalanobis distance.

If `Plot=T`, a plot of the instances ranked by their Mahalanobis distance is provided.

Author(s)

Edgar Acuna

References

Rousseeuw, P, and Leroy, A. (1987). Robust Regression and outlier detection. John Wiley & Sons. New York.

See Also

[robout](#)

Examples

```
#---- Detecting outliers using the Mahalanobis distance----  
data(bupa)  
mahaout(bupa,1)
```

mardia

The Mardia's test of normality

Description

Performs the Mardia's test to check for multivariate normality

Usage

```
mardia(data)
```

Arguments

data	The dataset containing the features for which multivariate normality is going to be tested. The last column contains the class. In case of unsupervised data add a dummy column of ones. In case of regression data, transform the response column in a column of ones
------	--

Value

Returns the p-values for the corresponding third and fourth moments of the multivariate normal distribution.

Author(s)

Edgar Acuna

References

Mardia, K.V. (1985). "Mardia's Test of Multinormality," in S. Kotz and N.L. Johnson, eds., Encyclopedia of Statistical Sciences, vol. 5 (NY: Wiley), pp. 217-221.

See Also

[vvalen](#)

Examples

```
#-----Mardia test for supervised data-----
data(my.iris)
mardia(my.iris)
#----Mardia test for unsupervised data-----
data(hawkins)
haw=cbind(hawkins[, -4], rep(1, 75))
mardia(haw)
```

maxdist	<i>Auxiliary function used when executing the Bay's algorithm for outlier detection</i>
---------	---

Description

This function is used by the function baysout in this package, to find the largest value of a distance vector. Returns the value and the index number of the largest distance.

Usage

```
maxdist(dneighbors)
```

Arguments

dneighbors The value and the index number of the largest distance

Author(s)

Caroline Rodriguez

See Also

[baysout](#)

maxlof	<i>Detection of multivariate outliers using the LOF algorithm</i>
--------	---

Description

A function that detects multivariate outliers using the local outlier factor for a matrix over a range of neighbors called minpts.

Usage

```
maxlof(data, name = "", minpts1 = 10, minptsu = 20)
```

Arguments

data	Dataset for outlier detection
name	Name of dataset used in the graph title.
minptsl	Lower bound for the number of neighbors
minptsu	Upper bound for the number of neighbors

Details

Calls on the function "lofactor" to compute the local outlier factor for each integer number of neighbors in the range [minptsl, minptsu]. Also displays a plot of the factors for each observation of the dataset. In the plot, the user should seek to identify observations with large gaps between outlyingness measures. These would be candidates for outliers.

Value

maxlofactor A vector containing the index of each observation of the dataset and the corresponding local outlier factor.

Author(s)

Caroline Rodriguez

References

Breuning, M., Kriegel, H., Ng, R.T, and Sander. J. (2000). LOF: Identifying density-based local outliers. In Proceedings of the ACM SIGMOD International Conference on Management of Data.

Examples

```
#Detecting top 10 outliers in class number 1 of Breastw using the LOF algorithm
data(breastw)
breastw1.lof=maxlof(breastw[breastw[,10]==1,],name="Breast-Wisconsin",30,40)
breastw1.lof[order(breastw1.lof,decreasing=TRUE)][1:10]
```

midpoints

Auxiliary function for computing minimum entropy discretization

Description

This function carries out the partial computation of the minimum entropy discretization

Usage

```
midpoints(x)
```

Arguments

x A numerical vector

Author(s)

Luis Daza

See Also

[disc.mentr](#)

 mmnorm

Min-max normalization

Description

This is a function to apply min-max normalization to a matrix or dataframe.

Usage

```
mmnorm(data, minval=0, maxval=1)
```

Arguments

data	The dataset to be normalized, including classes
minval	The minimum value of the transformed range
maxval	The maximum value of the transformed range

Details

Min-max normalization subtracts the minimum value of an attribute from each value of the attribute and then divides the difference by the range of the attribute. These new values are multiplied by the new range of the attribute and finally added to the new minimum value of the attribute. These operations transform the data into a new range, generally [0,1]. The function removes classes (assuming they are in last column) before normalization, and returns a normalized data set, complete with classes. Uses the function scale from the base package.

Value

zdata3	The normalized dataset
--------	------------------------

Author(s)

Caroline Rodriguez and Edgar Acuna

References

Hann, J., Kamber, M. (2000). Data Mining: Concepts and Techniques. Morgan Kaufman Publishers.

Examples

```
#---- Min-Max Normalization----
data(ionosphere)
ionos.minmax=mmnorm(ionosphere)
op=par(mfrow=c(2,1))
plot(ionosphere[,1])
plot(ionos.minmax[,1])
par(op)
```

`mo3`*The third moment of a multivariate distribution*

Description

This function computes the third moment of a multivariate normal distribution. This result is used later on the Mardia's test for multivariate normality

Usage

```
mo3(data)
```

Arguments

`data` The dataset containing the features of the multivariate vector for which the third moment will be computed

Value

`mo3` The third moment of the multivariate distribution

Author(s)

Edgar Acuna

See Also

[mo4](#), [mardia](#)

Examples

```
data(my.iris)
mo3(my.iris)
```

`mo4`*The fourth moment of a multivariate distribution*

Description

This function computes the fourth moment of a multivariate distribution. This result is used later in the mardia's test for multivariate normality.

Usage

```
mo4(data)
```

Arguments

`data` The dataset containing the features of the multivariate vector for which the fourth moment will be computed

Value

Returns the fourth moment.

Author(s)

Edgar Acuna

See Also

[mo3](#), [mardia](#)

Examples

```
data(my.iris)
mo4(my.iris)
```

moda

Calculating the Mode

Description

This function calculates the mode of a vector.

Usage

```
moda(x, na.rm = TRUE)
```

Arguments

x	A numeric vector
na.rm	A Boolean value that indicates the presence of missing values.

Details

The function returns the mode or modes of a vector. If a tie exists, all values that are tied are returned.

Value

moda A numeric value representing the mode of the vector

Author(s)

Caroline Rodriguez and Edgar Acuna

Examples

```
#---- Calculating the mode ----
x=c(1,4,2,3,4,6,3,7,8,5,4,3)
moda(x)
```

`my.iris`*The Iris dataset*

Description

This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. (See Duda & Hart, for example.) The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. The Setosa class is linearly separable from the other two classes. The last two classes are NOT linearly separable from each other.

Usage

```
data(my.iris)
```

Format

A dataframe with 150 observations on the following 5 variables.

V1 sepal length in cm

V2 sepal width in cm

V3 petal length in cm

V4 petal width in cm

V5 class: Iris Setosa(1), Iris Versicolor(2),Iris Virginica(3)

Source

The UCI Machine Learning Database Repository at:

- <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>
- <http://www.ics.uci.edu/~mllearn/MLRepository.html>

References

Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, Vol 7, Part II, 179–188.

Examples

```
#----Testing multivariate normality----  
data(my.iris)  
mardia(my.iris)
```

near1 *Auxiliary function for the reliefcont function*

Description

This function finds the instance in the data matrix that is closest to a given instance x. It is assumed that all the attributes are continuous.

Usage

```
near1(x, data)
```

Arguments

x	A given instance
data	The name of the dataset

Author(s)

Edgar Acuna

See Also

[near2,relief](#)

near2 *Auxiliary function for the reliefcat function*

Description

This function finds the instance in the data matrix that is closest to a given instance x. The attributes can be either continuous or nominal.

Usage

```
near2(x, data, vnom)
```

Arguments

x	A given instance
data	The name of the dataset
vnom	A vector of indexes for nominal attributes

Author(s)

Edgar Acuna

See Also

[relief,near1](#)

`nnmiss`*Auxiliary function for knn imputation*

Description

This function is required to perform k-nn imputation.

Usage

```
nnmiss(x, xmiss, ismiss, xnom, K = 1)
```

Arguments

<code>x</code>	A submatrix of complete rows from original matrix
<code>xmiss</code>	A a row with a missing value
<code>ismiss</code>	A vector that indicates whether a value in <code>xmiss</code> is missing or not
<code>xnom</code>	A vector with indexes of nominal variables
<code>K</code>	The number of neighbors to use

Author(s)

Edgar Acuna

See Also

[ce.impute](#)

`outbox`*Detecting outliers through boxplots of the features.*

Description

This function detects univariate outliers simultaneously using boxplots of the features.

Usage

```
outbox(data, nclass)
```

Arguments

<code>data</code>	The dataset to be explored for outlier detection.
<code>nclass</code>	A value representing the class that will be explored.

Details

The function also displays a plot containing a boxplot for of the variables.

Value

out1 A list of the indices of the observations that are outside the extremes of the boxplot. The indices are given in a table format representing the number of columns in which the observation was identified as an outlier.

Author(s)

Edgar Acuna

Examples

```
#---- Identifying outliers in diabetes-class1 with boxplots----
data(diabetes)
outbox(diabetes,nclass=1)
```

parallelplot *Parallel Coordinate Plot*

Description

Constructs a parallel coordinate plot for a data set with classes in last column.

Usage

```
parallelplot(x, name = "", comb = -1, class = 0, obs = rep(0, 0), col = 2, lty =
```

Arguments

x	A matrix of numerical values with classes in last column
name	The name of data set as will appear in the graph title
comb	An integer that represents the number of one of the possible combinations for the columns of this matrix.
class	A value representing the class number to which the plot should be limited
obs	A list of one or more row numbers that are to be highlighted in the plot
col	A value that provides a choice of color for the plot (if plotting only one class)
lty	A value that provides a choice of line width for the plot (if plotting only one class)
...	Additional arguments for the matplot function

Details

This plot is not recommended for a large number of features (say more than 50). If comb=0, all distinct combinations of columns are graphed. If comb=-1 (default), the attributes are plotted in their original order, else comb should be equal to an integer that represents the number of one of the possible combinations for the columns of this matrix.

Value

A parallel coordinate plot of the data is produced.

Author(s)

Caroline Rodriguez

References

Wegman, E. (1990), Hyperdimensional data analysis using parallel coordinates, *Journal of the American Statistical Association*, 85, 664-675

See Also

[starcoord](#), [surveyplot](#)

Examples

```
#---Parallel Coordinate Plot---  
data(bupa)  
parallelplot(bupa, "Bupa Dataset")  
parallelplot(bupa, "Bupa Dataset", comb=0)  
#parallelplot(bupa, "Bupa Dataset", comb=1, c(1, 22, 50))
```

pp.golub

The preprocessed Golub's dataset

Description

This is the preprocessed Golub's dataset, where the training and the learning set are first joined and then preprocessed according to the Dudoit et al.'s paper.

Usage

```
data(pp.golub)
```

Format

A data frame with 72 observations on 3572 variables.

Source

This data set is available at:

- <http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi>
- <http://www.bioconductor.org>

References

Dudoit, S., Fridlyand, J. & Speed, T. P. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *J Am Stat Assoc*, 97 (457), 77-87.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. & Lander, E. S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286, 531-537.

Examples

```
#----z-score Normalization---
data(pp.golub)
rangennorm(pp.golub, "znorm")
```

radviz2d

Radial Coordinate Visualization

Description

Radviz is a radial spring-based visualization that permits the visualization of n-dimensional datasets. Data attributes are equidistantly distributed along the circumference of a circle. Each data item is virtually connected to a spring that starts at the circle perimeter and ends on the data item. Each spring pulls the item with a force proportional to the item's attribute value. Depending on the value of each attribute, the forces of the springs project each data item to a position inside the circle where the sum of the spring forces is equal to zero.

Usage

```
radviz2d(dataset, name = "")
```

Arguments

dataset	The dataset to be visualized.
name	The name of the dataset to be used in the graph title.

Details

Some features of this visualization are: 1) Points where all dimensional values have approximately the same value will lie close to the center. 2) If dimensional points lie opposite each other on the circle and have similar values than points will lie near the center. 3) If 1 or 2 dimensional values are greater, points will lie closer to those dimensional points. 4) Where a point will lie depends on the layout of the particular dimensions around the circle. 5) This is a non-linear projection from N-dimensions down to 2 dimensions 6) Certain symmetries of the data will be preserved.

The function assumes the class labels are in the last column. Class column may be either a numeric vector or a factor.

Value

A Radviz visualization of the original dataset is returned.

Note

Prior to visualizing, the values of each attribute are usually standardized to the interval [0, 1] to make all the attributes equally important in "pulling" the data point. If one attribute value is much larger than the values of the other attributes, then the point will lie close to the point on the circumference of the circle which corresponds to this attribute. The visualization of a given data set, and also its usefulness, largely depends on the selection of visualized attributes and their ordering around the circle perimeter. The total number of possible orderings of m attributes is factorial(m), but some of them are equivalent up to a rotation or image mirroring. Hence, it can be shown that the total number of different projections with m attributes is factorial(m-1)/2.

Author(s)

Caroline Rodriguez

References

Ankerst M., Keim D. A., Kriegel H.-P. Circle Segments: A Technique for Visually Exploring Large Multidimensional Data Sets, IEEE Visualization, 1996.

K.A. Olsen, R.R. Korfhage, K.M. Sochats, M.B. Spring and J.G. Williams. Visualisation of a Document Collection: The VIBE System, Information Processing and Management, Vol. 29, No. 1, pp. 69-81, Pergamon Press Ltd, 1993.

See Also

[starcoord](#), [surveyplot](#), [paralleplot](#)

Examples

```
data(my.iris)
radviz2d(my.iris, "Iris")
```

rangenorm

range normalization

Description

Performs several methods of range normalization.

Usage

```
rangenorm(data, method = c("znorm", "mmnorm", "decscale", "signorm", "softmaxnorm"))
```

Arguments

`data` The name of the dataset to be normalized

`method` The discretization method to be used: "znorm", "mmnorm", "decscale", "signorm", "softmaxnorm"

`superv`

superv=T for supervised data, that data including the class labels in the last column. if superv=F means that the data to be used is unsupervised.

Details

In the znorm normalization, the mean of each attribute of the transformed set of data points is reduced to zero by subtracting the mean of each attribute from the values of the attributes and dividing the difference by the standard deviation of the attribute. Uses the function scale found in the base library.

Min-max normalization (mmnorm) subtracts the minimum value of an attribute from each value of the attribute and then divides the difference by the range of the attribute. These new values are multiplied by the new range of the attribute and finally added to the new minimum value of the attribute. These operations transform the data into a new range, generally [0,1].

The decscale normalization applies decimal scaling to a matrix or dataframe. Decimal scaling transforms the data into $[-1,1]$ by finding k such that the absolute value of the maximum value of each attribute divided by 10^k is less than or equal to 1.

In the sigmoidal normalization (signorm) the input data is nonlinearly transformed into $[-1,1]$ using a sigmoid function. The original data is first centered about the mean, and then mapped to the almost linear region of the sigmoid. Is especially appropriate when outlying values are present.

The softmax normalization is so called because it reaches "softly" towards maximum and minimum value, never quite getting there. The transformation is more or less linear in the middle range, and has a nonlinearity at both ends. The output range covered is $[0,1]$. The algorithm removes the classes of the dataset before normalization and replaces them at the end to form the matrix again.

Value

A matrix containing the discretized data.

Author(s)

Caroline Rodriguez and Edgar Acuna

References

Caroline Rodriguez (2004). An computational environment for data preprocessing in supervised classification. Master thesis. UPR-Mayaguez

Hann, J., Kamber, M. (2000). Data Mining: Concepts and Techniques. Morgan Kaufman Publishers.

Examples

```
#----Several methods of range normalization ----
data(bupa)
bupa.znorm=rangenorm(bupa,method="znorm")
bupa.mmnorm=rangenorm(bupa,method="mmnorm")
bupa.decs=rangenorm(bupa,method="decscale")
bupa.signorm=rangenorm(bupa,method="signorm")
bupa.soft=rangenorm(bupa,method="softmaxnorm")
#----Plotting to see the effect of the normalization----
op=par(mfrow=c(2,3))
plot(bupa[,1])
plot(bupa.znorm[,1])
plot(bupa.mmnorm[,1])
plot(bupa.decs[,1])
plot(bupa.signorm[,1])
plot(bupa.soft[,1])
par(op)
```

reachability

Function for computing the reachability measure in the LOF algorithm

Description

This function computes the reachability measure for each instance of a dataset. This result is used later to compute the Local Outlyingness Factor.

Usage

```
reachability(distdata, k)
```

Arguments

distdata	The matrix of distances
k	The given number of neighbors

Author(s)

Caroline Rodriguez

See Also

[maxlof](#)

redundancy	<i>Finding the unique observations in a dataset along with their frequencies</i>
------------	--

Description

This function finds out the unique instances in a dataset along with their frequencies.

Usage

```
redundancy(data)
```

Arguments

data	The name of the dataset
------	-------------------------

Author(s)

Edgar Acuna

See Also

[clean](#)

`relief`*RELIEF Feature Selection*

Description

This function implements the RELIEF feature selection algorithm.

Usage

```
relief(data, nosample, threshold, vnom)
```

Arguments

<code>data</code>	The dataset for which feature selection will be carried out
<code>nosample</code>	The number of instances drawn from the original dataset
<code>threshold</code>	The cutoff point to select the features
<code>vnom</code>	A vector containing the indexes of the nominal features

Details

The general idea of this method is to choose the features that can be most distinguished between classes. These are known as the relevant features. At each step of an iterative process, an instance x is chosen at random from the dataset and the weight for each feature is updated according to the distance of x to its Nearmiss and NearHit.

Value

<code>relevant</code>	A table that gives the ratio between the frequency with which the feature was selected as relevant and the total number of trials performed in one column, and the average weight of the feature in another.
<code>a plot</code>	A plot of the weights of the features

Author(s)

Edgar Acuna

References

KIRA, K. and RENDEL, L. (1992). The Feature Selection Problem : Traditional Methods and a new algorithm. Proc. Tenth National Conference on Artificial Intelligence, MIT Press, 129-134.

KONONENKO, I., SIMEC, E., and ROBNIK-SIKONJA, M. (1997). Overcoming the myopia of induction learning algorithms with RELIEFF. Applied Intelligence Vol7, 1, 39-55.

Examples

```
##---- Feature Selection ---  
data(my.iris)  
relief(my.iris,150,0.01)
```

reliefcat	<i>Feature selection by the Relief Algorithm for datasets with only nominal features</i>
-----------	--

Description

This function applies the RELIEF Algorithm to datasets containing nominal attributes.

Usage

```
reliefcat(data, nosample, threshold, vnom)
```

Arguments

data	The name of the dataset
nosample	The size of the sample drawn and used to update the relevance of each feature
threshold	The threshold for choosing the relevant features
vnom	A vector of indices indicating the nominal features

Author(s)

Edgar Acuna

See Also

[relief](#)

reliefcont	<i>Feature selection by the Relief Algorithm for datasets with only continuous features</i>
------------	---

Description

This function applies Relief to datasets containing only continuous attributes.

Usage

```
reliefcont(data, nosample, threshold)
```

Arguments

data	The name of the dataset
nosample	The size of the sample drawn and use to update the relevance of the features
threshold	The threshold for choosing the relevant features.

Author(s)

Edgar Acuna

See Also

[relief](#)

`robout`*Outlier Detection with Robust Mahalanobis distance*

Description

This function finds the outliers of a dataset using robust versions of the Mahalanobis distance.

Usage

```
robout(data, nclass, meth = c("mve", "mcd"), rep = 10,  
plot = TRUE)
```

Arguments

<code>data</code>	The dataset for which outlier detection will be carried out.
<code>nclass</code>	An integer value that represents the class to detect for outliers
<code>meth</code>	The method used to compute the Mahalanobis distance, "mve"=minimum volume estimator, "mcd"=minimum covariance determinant
<code>rep</code>	Number of repetitions
<code>plot</code>	A boolean value to turn on and off the scatter plot of the Mahalanobis distances

Details

Requires uses `cov.rob` function from the MASS library.

Value

<code>top1</code>	
<code>topout</code>	Index of observations identified as possible outliers by outlyingness measure
<code>outme</code>	Index of observations and their outlyingness measures

Author(s)

Edgar Acuna

References

Rousseeuw, P, and Leroy, A. (1987). Robust Regression and outlier detection. John Wiley & Sons. New York.

Atkinson, A. (1994). Fast very robust methods for the detection of multiple outliers. Journal of the American Statistical Association, 89:1329-1339.

Examples

```
#---- Outlier Detection in bupa-class 1 using MCD  
data(bupa)  
robout(bupa,1,"mcd")
```

row.matches *Finding rows in a matrix equal to a given vector*

Description

This function finds instances in a data matrix that are equal to a given instance.

Usage

```
row.matches(y, X)
```

Arguments

y	A given instance
X	A given data matrix

Details

This function was found in the CRAN mailing list. It seems to be authored by B. Venables

See Also

[redundancy](#)

sbs1 *One-step sequential backward selection*

Description

This functions performs one-step of the sequential backward selection procedure.

Usage

```
sbs1(data, indic, correct0, kvec, method = c("lda", "knn", "rpart"))
```

Arguments

data	The name of a dataset
indic	A vector of 0-1 values: 1 indicates a selected feature.
correct0	The recognition rate based on the current subset of features
kvec	The number of neighbors
method	The classifier to be used

Author(s)

Edgar Acuna

See Also

[sffs](#)

`score`*Score function used in Bay's algorithm for outlier detection*

Description

This function finds the score that is used to rank an instance as an outliers.

Usage

```
score(data)
```

Arguments

`data` The name of the dataset to be used.

Author(s)

Caroline Rodriguez

See Also

[baysout](#)

`sffs`*Sequential Floating Forward Method*

Description

This function selects features using the sequential floating forward method with lda, knn or rpart classifiers.

Usage

```
sffs(data, method = c("lda", "knn", "rpart"), kvec = 5,  
repet = 10)
```

Arguments

`data` Dataset to be used for feature selection
`method` String sequence representing the choice of classifier
`kvec` The number of nearest neighbors to be used for the knn classifier
`repet` Integer value representing the number of repetitions

Details

The Sequential Floating Forward selection method was introduced to deal with the nesting problem. The best subset of features, T, is initialized as the empty set and at each step a new feature is added. After that, the algorithm searches for features that can be removed from T until the correct classification error does not increase. This algorithm is a combination of the sequential forward and the sequential backward methods. The "best subset" of features is constructed based on the frequency with which each attribute is selected in the number of repetitions given. Due to the time complexity of the algorithm its use is not recommended for data sets with a large number of attributes (say more than 1000).

Value

`fselect` a list of the indices of the best features

Author(s)

Edgar Acuna

References

Pudil, P., Ferri, J., Novovicova, J., and Kittler, J. (1994). Floating search methods for feature selection with nonmonotonic criterion function. 12 International Conference on Pattern Recognition, 279-283.

Acuna, E , (2003) A comparison of filters and wrappers for feature selection in supervised classification. Proceedings of the Interface 2003 Computing Science and Statistics. Vol 34.

Examples

```
#---- SFFS feature selection using the knn classifier ----
data(my.iris)
sfs(my.iris,method="knn",kvec=5,repet=5)
```

sfs

Sequential Forward Selection

Description

Applies the Sequential Forward Selection algorithm for Feature Selection.

Usage

```
sfs(data, method = c("lda", "knn", "rpart"), kvec = 5,
     repet = 10)
```

Arguments

<code>data</code>	Dataset to be used for feature selection
<code>method</code>	Classifier to be used, currently only the lda, knn and rpart classifiers are supported
<code>kvec</code>	Number of neighbors to use for the knn classification
<code>repet</code>	Number of times to repeat the selection.

Details

The best subset of features, T, is initialized as the empty set and at each step the feature that gives the highest correct classification rate along with the features already in T, is added to set. The "best subset" of features is constructed based on the frequency with which each attribute is selected in the number of repetitions given. Due to the time complexity of the algorithm its use is not recommended for datasets with a large number of attributes(say more than 1000).

Value

bestsubset subset of features that have been determined to be relevant.

Author(s)

Edgar Acuna

References

Acuna, E , (2003) A comparison of filters and wrappers for feature selection in supervised classification. Proceedings of the Interface 2003 Computing Science and Statistics. Vol 34.

Examples

```
#---- Sequential forward selection using the knn classifier----
data(my.iris)
sfs(my.iris,method="knn",kvec=3,repet=10)
```

sfs1

One-step sequential forward selection

Description

This function computes one-step of the sequential forward selection procedure.

Usage

```
sfs1(data, indic, correcto, kvec, method = c("lda", "knn",
      "rpart"))
```

Arguments

data	Name of the dataset to be used.
indic	A vector of 0-1 values.
correcto	The recognition rate in the previous step.
kvec	The number of neighbors to be used by the knn classifier
method	The classifier to be used to select the best features.

Author(s)

Edgar Acuna

See Also

[sffs](#)

`signorm`*Sigmoidal Normalization*

Description

Function that performs sigmoidal normalization.

Usage

```
signorm(data)
```

Arguments

`data` The dataset to be normalized, including classes

Details

This method transforms the input data nonlinearly into $[-1,1]$ using a sigmoid function. The original data is first centered about the mean, and then mapped to the almost linear region of the sigmoid. Is especially appropriate when outlying values are present.

Removes classes before normalization, and returns the normalized data set complete with classes rejoined.

Value

`sigdata` Original dataset normalized

Author(s)

Caroline Rodriguez and Edgar Acuna

References

Hann, J., Kamber, M. (2000). Data Mining: Concepts and Techniques. Morgan Kaufman Publishers.

Examples

```
#---- Sigmoidal Normalization ---
data(vehicle)
vehicle$signorm=signorm(vehicle)
op=par(mfrow=c(2,1))
plot(vehicle[,1])
plot(vehicle$signorm[,1])
par(op)
```

softmaxnorm	<i>Softmax Normalization</i>
-------------	------------------------------

Description

This is a function that applies softmax normalization to a matrix or dataframe.

Usage

```
softmaxnorm(data)
```

Arguments

data	The dataset to be normalized
------	------------------------------

Details

This normalization is so called because it reaches "softly" towards maximum and minimum value, never quite getting there. The transformation is more or less linear in the middle range, and has a nonlinearity at both ends. The output range covered is [0,1]. The algorithm removes the classes of the dataset before normalization and replaces them at the end to form the matrix again.

Value

softdata	original matrix normalized
----------	----------------------------

Author(s)

Caroline Rodriguez and Edgar Acuna

Examples

```
#---- Softmax Normalization----  
data(sonar)  
sonar.sftnorm=softmaxnorm(sonar)  
op=par(mfrow=c(2,1))  
plot(sonar[,1])  
plot(sonar.sftnorm[,1])  
par(op)
```

sonar	<i>The Sonar dataset</i>
-------	--------------------------

Description

This is the sonar dataset. It contains information on 208 objects and 60 attributes. The objects are classified in two classes: "rock" and "mine".

Usage

```
data(sonar)
```

Format

A data frame with 208 observations on 61 variables. The first 60 represent the energy within a particular frequency band, integrated over a certain period of time. The last column contains the class labels. There are two classes 0 if the object is a rock, and 1 if the object is a mine (metal cylinder). The range value of each attribute varies from 0.0 to 1.0.

Source

The UCI Machine Learning Database Repository at:

- <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>
- <http://www.ics.uci.edu/~mllearn/MLRepository.html>

Examples

```
#Robust detection of outliers in sonar-class1 using MVE----  
data(sonar)  
robout(sonar, 1, "mve", rep=10)
```

srbct

Khan et al.'s small round blood cells dataset

Description

The srbct dataset which contains information on 63 samples and 2308 genes. The samples are distributed in four classes as follows: 8 Burkitt Lymphoma (BL), 23 Ewing Sarcoma (EWS), 12 neuroblastoma (NB), and 20 rhabdomyosarcoma (RMS). The last column contains the class labels.

Usage

```
data(srbct)
```

Format

A data frame containing 63 observations with 2308 attributes each. The last column of the data frame contains the class labels for each observation.

Source

The data set was obtained, as binary R file from Marcel Dettling's web site:

- <http://stat.ethz.ch/~dettling/bagboost.html>

References

Javed Khan, Jun S. Wei, Markus Ringner, Lao H. Saal, Marc Ladanyi, Frank Westermann, Frank Berthold, Manfred Schwab, Cristina R. Antonescu, Carsten Peterson, and Paul S. Meltzer (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, Volume 7, Number 6, June

Examples

```
#---z-score Normalization
data(srbct)
srbct.rnorm=rangenorm(srbct,"znorm")
#---feature selection using the RELIEF feature selection algorithm-----
#relief(srbct,63,0.12)
```

starcoord

The star coordinates plot

Description

This function displays a star coordinates plot introduced by Kondogan (2001).

Usage

```
starcoord(data, main = NULL, class = FALSE, outliers=NULL, vars = 0,
scale = 1, cex = 0.8, lwd = 0.25, lty = par("lty"))
```

Arguments

data	The dataset
main	The title of the plot
class	This logical variable is TRUE for supervised data and FALSE for unsupervised data
outliers	The instances to be highlighted as potential outliers
vars	The variables to be scaled
scale	The scale factor
cex	A numerical value giving the amount by which plotting text and symbols should be scaled.
lwd	The width of the lines representing the axis
lty	The type of the lines representing the axis

Details

This plot is not recommended for a large number of features (say more than 50).

Value

Returns a Star Coordinates Plot of the data matrix

Author(s)

Edgar Acuna and Shiyun Wen

References

E. Kandogan (2001). Visualizing multidimensional clusters, Trends, and Outliers, using star coordinates. Proceedings of KDD 2001.

See Also

[parallelplot](#), [surveyplot](#)

Examples

```
data(bupa)
starcoord(bupa, main="Bupa Dataset", class=TRUE, outliers=NULL, vars=0, scale=1,
cex=0.8, lwd = 0.25, lty = par("lty"))
```

surveyplot

Surveyplot

Description

This function creates and displays a surveyplot of a dataset for a classification matrix

Usage

```
surveyplot(datos, dataname = "", orderon = 0, class = 0,
obs = rep(0, 0), lwd = 1)
```

Arguments

datos	A matrix of values for supervised classification
dataname	dataname Name of data set to appear in plot title
orderon	orderon Column number by which to order the dataset
class	class Class for which to limit plotting
obs	obs List of observations to be highlighted
lwd	lwd Value to control width of the line

Details

This plot is not recommended for a large number of features (say more than 50)

Value

Returns a surveyplot of the data matrix

Note

This plot is a mix between the survey plot presented in Fayyad and a permutation matrix.

Author(s)

Caroline Rodriguez

References

Fayyad, et al. (2001) Information Visualization in Data Mining and Knowledge Discovery

See Also

[parallelplot](#), [starcoord](#)

Examples

```
#----Surveyplot examples
data(bupa)
surveyplot(bupa, "Bupa Dataset")
surveyplot(bupa, "Bupa Dataset", orderon=1, obs=c(6, 74, 121))
```

tchisq

Auxiliary function for the Chi-Merge discretization

Description

This function is required to compute the chi-Merge discretization.

Usage

```
tchisq(obs)
```

Arguments

obs a vector of observed frequencies

Author(s)

Jaime Porras

See Also

[chiMerge](#)

top

Auxiliary function for Bay's Ouylier Detection Algorithm

Description

Function that finds the number of candidate outliers requested by the user.

Usage

```
top(O, neighbors, n)
```

Arguments

O An n x 1 matrix with the score function from k nearest neighbors
neighbors The number of neighbors to be considered
n The number of top outliers to search for.

Author(s)

Caroline Rodriguez

See Also

[baysout](#)

vehicle

The Vehicle dataset

Description

This is the Vehicle dataset from the UCI Machine Learning Repository

Usage

```
data(vehicle)
```

Format

A data frame with 846 observations on the following 19 variables.

- V1** Compactness
- V2** Circularity
- V3** Distance Circularity
- V4** Radius ratio
- V5** pr.axis aspect ratio
- V6** max.length aspect ratio
- V7** scatter ratio
- V8** elongatedness
- V9** pr.axis rectangularity
- V10** max.length rectangularity
- V11** scaled variance along major axis
- V12** scaled variance along minor axis
- V13** scaled radius of gyration
- V14** skewness about major axis
- V15** skewness about minor axis
- V16** kurtosis about minor axis
- V17** kurtosis about major axis
- V18** hollows ratio
- V19** Type of vehicle: a double decker bus, Cheverolet van, Saab 9000 and an Opel Manta 400.

Source

The UCI Machine Learning Database Repository at:

- <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>
- <http://www.ics.uci.edu/~mllearn/MLRepository.html>

Examples

```
#----feature selection using sequential floating selection with LDA----  
data(vehicle)  
mahaout(vehicle,nclass=3)
```

vvalen

The Van Valen test for equal covariance matrices

Description

The Van Valen nonparametric test for homocedasticity (equal covariance matrices).

Usage

```
vvalen(data)
```

Arguments

data The name of the dataset to be tested

Value

Gives the p-value for a Kruskal Wallis test. A p-value near to zero indicates homocedasticity.

Author(s)

Edgar Acuna

References

Van Valen, L. (1962). A study of fluctuating asymmetry. *Evolution* Vol. 16, pp. 125-142.

See Also

[mardia](#)

Examples

```
#-----Testing homocedasticity-----  
data(colon)  
vvalen(colon)
```

`vvalen1`*Auxiliary function for computing the Van Valen's homocedasticity test*

Description

This function is required to perform the Van Valen's homocedasticity test.

Usage

```
vvalen1(data, classn)
```

Arguments

<code>data</code>	The name of the dataset to be considered
<code>classn</code>	The class number

Author(s)

Edgar Acuna

See Also

[vvalen](#)

`znorm`*Z-score normalization*

Description

This is a function to apply z-Score normalization to a matrix or dataframe.

Usage

```
znorm(data)
```

Arguments

<code>data</code>	The dataset to be normalized, including classes
-------------------	---

Details

By using this type of normalization, the mean of the transformed set of data points is reduced to zero by subtracting the mean of each attribute from the values of the attributes and dividing the result by the standard deviation of the attribute. Uses the function `scale` found in the base library.

Removes classes before normalization, and returns normalized data set complete with classes re-joined.

Value

<code>zdata</code>	the normalized data set
--------------------	-------------------------

Author(s)

Caroline Rodriguez and Edgar Acuna

References

Hann, J., Kamber, M. (2000). Data Mining: Concepts and Techniques. Morgan Kaufman Publishers.

Examples

```
##---- Z-norm normalization ----  
data(diabetes)  
diab.znorm=znorm(diabetes)  
op=par(mfrow=c(2,1))  
plot(diabetes[,1])  
plot(diab.znorm[,1])  
par(op)
```

Index

*Topic **classif**

- crossval, 14
- cv10knn2, 15
- cv10lda2, 15
- cv10log, 16
- cv10mlp, 17
- cv10rpart2, 18

*Topic **datasets**

- breastw, 3
- bupa, 4
- censusn, 8
- colon, 12
- diabetes, 19
- ejeldis, 27
- hawkins, 28
- heartc, 29
- hepatitis, 30
- ionosphere, 33
- my.iris, 43
- pp.golub, 47
- sonar, 60
- srbc, 61
- vehicle, 65

*Topic **hplot**

- circledraw, 10
- imagmiss, 31
- parallelplot, 46
- radviz2d, 48
- starcoord, 62
- surveyplot, 63

*Topic **manip**

- ce.impute, 5
- ce.knn.imp, 6
- ce.mimp, 7
- chiMerge, 9
- clean, 11
- combinations, 13
- decscale, 18
- disc.lr, 20
- disc.ef, 21
- disc.ew, 22
- disc.mentr, 22
- disc2, 23

- discretevar, 24
- ec.knnimp, 26
- mmnorm, 40
- nnmiss, 45
- rangenorm, 49
- redundancy, 51
- row.matches, 55
- signorm, 59
- softmaxnorm, 60
- znorm, 67

*Topic **math**

- assig, 2
- closest, 12
- dist.to.knn, 24
- distan2, 25
- distancia, 25
- kneigh.vect, 34
- maxdist, 38
- midpoints, 39
- near1, 44
- near2, 44
- reachability, 50
- score, 56
- tchisq, 64

*Topic **methods**

- baysout, 2
- finco, 27
- lofactor, 34
- lvf, 35
- mahaout, 36
- maxlof, 38
- outbox, 45
- relief, 52
- reliefcat, 53
- reliefcont, 53
- robout, 54
- sbs1, 55
- sffs, 56
- sfs, 57
- sfs1, 58
- top, 64
- vvalen1, 67

*Topic **misc**

- inconsist, [32](#)
- vvalen, [66](#)
- *Topic multivariate**
 - mardia, [37](#)
 - mo3, [41](#)
 - mo4, [41](#)
- *Topic package**
 - dprep-package, [1](#)
- *Topic univar**
 - moda, [42](#)
- assig, [2](#)
- baysout, [2](#), [12](#), [38](#), [56](#), [65](#)
- breastw, [3](#)
- bupa, [4](#)
- ce.impute, [5](#), [11](#), [45](#)
- ce.knn.imp, [6](#)
- ce.mimp, [7](#)
- censusn, [8](#)
- chiMerge, [9](#), [20–23](#), [64](#)
- circledraw, [10](#)
- clean, [5](#), [11](#), [51](#)
- closest, [12](#)
- colon, [12](#)
- combinations, [13](#)
- crossval, [14](#), [15–18](#)
- cv10knn2, [15](#)
- cv10lda2, [15](#)
- cv10log, [14](#), [16](#), [17](#)
- cv10mlp, [14](#), [16](#), [17](#)
- cv10rpart2, [18](#)
- decscale, [18](#)
- diabetes, [19](#)
- disc.1r, [9](#), [20](#), [21–23](#)
- disc.ef, [9](#), [20](#), [21](#), [22](#), [23](#)
- disc.ew, [9](#), [20](#), [21](#), [22](#), [23](#), [36](#)
- disc.mentr, [2](#), [9](#), [20](#), [22](#), [22](#), [24](#), [40](#)
- disc2, [23](#)
- discretevar, [24](#)
- dist.to.knn, [24](#)
- distan2, [25](#)
- distancia, [25](#)
- dprep (*dprep-package*), [1](#)
- dprep-package, [1](#)
- ec.knnimp, [26](#)
- ejeldis, [27](#)
- finco, [27](#), [33](#), [36](#)
- hawkins, [28](#)
- heartc, [29](#)
- hepatitis, [30](#)
- imagmiss, [31](#)
- inconsist, [28](#), [32](#), [36](#)
- ionosphere, [33](#)
- knneigh.vect, [34](#)
- lofactor, [34](#)
- lvf, [28](#), [33](#), [35](#)
- mahaout, [36](#)
- mardia, [37](#), [41](#), [42](#), [66](#)
- maxdist, [38](#)
- maxlof, [24](#), [34](#), [38](#), [51](#)
- midpoints, [39](#)
- mmnorm, [40](#)
- mo3, [41](#), [42](#)
- mo4, [41](#), [41](#)
- moda, [42](#)
- my.iris, [43](#)
- near1, [44](#), [44](#)
- near2, [44](#), [44](#)
- nnmiss, [45](#)
- outbox, [45](#)
- parallelplot, [46](#), [49](#), [63](#), [64](#)
- pp.golub, [47](#)
- radviz2d, [48](#)
- rangenorm, [49](#)
- reachability, [50](#)
- redundancy, [51](#), [55](#)
- relief, [25](#), [44](#), [52](#), [53](#)
- reliefcat, [53](#)
- reliefcont, [53](#)
- robout, [37](#), [54](#)
- row.matches, [55](#)
- sbs1, [55](#)
- score, [56](#)
- sffs, [55](#), [56](#), [58](#)
- sfs, [57](#)
- sfs1, [58](#)
- signorm, [59](#)
- softmaxnorm, [60](#)
- sonar, [60](#)
- srbct, [61](#)
- starcoord, [47](#), [49](#), [62](#), [64](#)
- surveyplot, [47](#), [49](#), [63](#), [63](#)
- tchisq, [64](#)

top, [64](#)

vehicle, [65](#)

vvalen, [37](#), [66](#), [67](#)

vvalen1, [67](#)

znorm, [67](#)