Package ‘lmap’

January 24, 2025

Type Package

Title Logistic Mapping

Version 0.2.4

Maintainer Mark de Rooij `<rooijm@fsw.leidenuniv.nl>`

Description Set of tools for mapping of categorical response variables based on principal component analysis (pca) and multidimensional unfolding (mdu).

Depends R (>= 3.5.0), ggplot2, ggrepel, ggforce, fmdu

Imports nnet, stats, magrittr, dplyr, MASS, Rfast, ggpubr, haven

License BSD_2_clause + file LICENSE

Encoding UTF-8

LazyData TRUE

Author Mark de Rooij [aut, cre, cph],
Frank Busing [aut, cph],
Juan Claramunt Gonzalez [aut]

NeedsCompilation yes

RoxygenNote 7.3.2

Repository CRAN

Date/Publication 2025-01-24 10:20:02 UTC

Contents

bootstrap.clmdu 3
bootstrap.clpca 4
bootstrap.lmdu 4
bootstrap.lpca 5
bootstrap.mcd 6
bootstrap.mrrr 7
bootstrap.mru 7
clmdu 8
clpca 10
dataExample_clmdu 11

dataExample_clpca . . . 12
dataExample_lmdu . . . 12
dataExample_lpca . . . 13
dataExample_mru . . . 14
diabetes . . . 14
dpes . . . 15
fastmbu . . . 15
fastmru . . . 17
kieskompas . . . 18
liver . . . 19
lmdu . . . 19
lpca . . . 21
make.df.for.varlabels . . . 23
make.dfs.for.X . . . 23
mcd1 . . . 24
mcd2 . . . 26
mlr . . . 27
mrrr . . . 28
mru . . . 29
nesda . . . 30
oos.comparison . . . 31
plot.bootstrap . . . 32
plot.clmdu . . . 32
plot.clpca . . . 34
plot.lmdu . . . 35
plot.lpca . . . 36
plot.mrrr . . . 37
plot.mru . . . 38
plot.trioscale . . . 39
predict.clmdu . . . 40
predict.clpca . . . 41
predict.lmdu . . . 42
predict.lpca . . . 43
predict.mlr . . . 44
predict.mrrr . . . 45
predict.mru . . . 46
procrustes1 . . . 47
procx . . . 47
read_drugdata . . . 48
read_isspdata_peb . . . 49
summary.clmdu . . . 49
summary.clpca . . . 50
summary.lmdu . . . 50
summary.lpca . . . 51
summary.mcd . . . 51
summary.mlr . . . 52
summary.mrrr . . . 52
summary.mru . . . 53

summary.trioscale 53
theme_lmda 54
trioscale 54
twomodedistance 55

Index **56**

`bootstrap.clmdu` *Bootstrap procedure for Cumulative Logistic (Restricted) MDU*

Description

Bootstrap procedure for Cumulative Logistic (Restricted) MDU

Usage

```
bootstrap.clmdu(object, Bsamples = 1000, myseed = 1)
```

Arguments

`object`	An output object from clmdu
`Bsamples`	Number of Bootstrap samples to take
`myseed`	A seed number to make the bootstrap reproducible

Value

BBdf Bootstrap estimates of B

BVdf Bootstrap estimates of V

Examples

```
## Not run:
data(dataExample_lmdu)
Y = as.matrix(dataExample_clmdu[ , 1:8])
X = as.matrix(dataExample_clmdu[ , 9:13])
output2 = clmdu(Y = Y, X = X, S = 2)
boot.output = bootstrap.lmdu(output2, Bsamples = 100)
plot(boot.output)

## End(Not run)
```

`bootstrap.clpca` *Bootstrap procedure for Cumulative Logistic (Restricted) PCA*

Description

Bootstrap procedure for Cumulative Logistic (Restricted) PCA

Usage

```
bootstrap.clpca(object, Bsamples = 1000, myseed = 1)
```

Arguments

`object`	An output object from clpca
`Bsamples`	Number of Bootstrap samples to take
`myseed`	A seed number to make the bootstrap reproducible

Value

BBdf Bootstrap estimates of B

BVdf Bootstrap estimates of V

Examples

```
## Not run:
data(dataExample_clpca)
Y<-as.matrix(dataExample_clpca[,5:8])
X<-as.matrix(dataExample_clpca[,1:4])
# supervised
output = clpca(Y = Y, X = X, S = 2)
boot.output = bootstrap.clpca(output, Bsamples = 100)
plot(boot.output)

## End(Not run)
```

`bootstrap.lmdu` *Bootstrap procedure for Logistic (Restricted) MDU*

Description

Bootstrap procedure for Logistic (Restricted) MDU

Usage

```
bootstrap.lmdu(object, Bsamples = 1000, myseed = 1)
```

Arguments

`object`	An output object from lmdu
`Bsamples`	Number of Bootstrap samples to take
`myseed`	A seed number to make the bootstrap reproducible

Value

BBdf Bootstrap estimates of B

BVdf Bootstrap estimates of V

Examples

```
## Not run:
data(dataExample_lmdu)
Y = as.matrix(dataExample_lmdu[ , 1:8])
X = as.matrix(dataExample_lmdu[ , 9:13])
output2 = lmdu(Y = Y, X = X, S = 2)
boot.output = bootstrap.lmdu(output2, Bsamples = 100)
plot(boot.output)

## End(Not run)
```

`bootstrap.lpca` *Bootstrap procedure for Logistic (Restricted) PCA*

Description

Bootstrap procedure for Logistic (Restricted) PCA

Usage

```
bootstrap.lpca(object, Bsamples = 1000, myseed = 1)
```

Arguments

`object`	An output object from lpca
`Bsamples`	Number of Bootstrap samples to take
`myseed`	A seed number to make the bootstrap reproducible

Value

BBdf Bootstrap estimates of B

BVdf Bootstrap estimates of V

Examples

```
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lpca[, 1:8])
X = as.matrix(dataExample_lpca[, 9:13])
# supervised
output = lpca(Y = Y, X = X, S = 2)
boot.output = bootstrap.lpca(output, Bsamples = 100)
plot(boot.output)

## End(Not run)
```

`bootstrap.mcd` *Bootstrap procedure for Multonimal Canonical Decomposition Model*

Description

Bootstrap procedure for Multonimal Canonical Decomposition Model

Usage

```
bootstrap.mcd(object, Bsamples = 1000)
```

Arguments

`object`	An output object from mcd1 or mcd2
`Bsamples`	Number of Bootstrap samples to take

Value

BBdf Bootstrap estimates of B

BVdf Bootstrap estimates of V

Examples

```
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lpca[, 1:8])
X = as.matrix(dataExample_lpca[, 9:13])
# supervised
output = mcd1(X, Y, S = 2, ord.z = 2)
#' boot.output = bootstrap.mcd(output, Bsamples = 100)
plot(boot.output)

## End(Not run)
```

`bootstrap.mrrr` *Bootstrap procedure for Multinomial Reduced Rank Model*

Description

Bootstrap procedure for Multinomial Reduced Rank Model

Usage

```
bootstrap.mrrr(object, Bsamples = 1000, myseed = 1)
```

Arguments

`object`	An output object from mrrr
`Bsamples`	Number of Bootstrap samples to take
`myseed`	A seed number to make the bootstrap reproducible

Value

BBdf Bootstrap estimates of B

BVdf Bootstrap estimates of V

Examples

```
## Not run:
data(dataExample_mru)
y = as.matrix(dataExample_mru[ , 1])
X = as.matrix(dataExample_mru[ , 2:6])
output = mrrr(y = y, X = X, S = 2)
boot.output = bootstrap.mrrr(output, Bsamples = 100)
plot(boot.output)

## End(Not run)
```

`bootstrap.mru` *Bootstrap procedure for Multinomial Restricted Unfolding*

Description

Bootstrap procedure for Multinomial Restricted Unfolding

Usage

```
bootstrap.mru(object, Bsamples = 1000, myseed = 1)
```

Arguments

`object`	An output object from mru
`Bsamples`	Number of Bootstrap samples to take
`myseed`	A seed number to make the bootstrap reproducible

Value

BBdf Bootstrap estimates of B

BVdf Bootstrap estimates of V

Examples

```
## Not run:
data(dataExample_mru)
y = as.matrix(dataExample_mru[ , 1])
X = as.matrix(dataExample_mru[ , 2:6])
output = mru(y = y, X = X, S = 2)
boot.output = bootstrap.mru(output, Bsamples = 100)
plot(boot.output)

## End(Not run)
```

`clmdu` *Cumulative Logistic (Restricted) MDU*

Description

Cumulative Logistic (Restricted) MDU

Usage

```
clmdu(
  Y,
  X = NULL,
  S = 2,
  trace = FALSE,
  start = "svd",
  maxiter = 65536,
  dcrit = 1e-06
)
```

Arguments

`Y`	An N times R ordinal matrix coded with integers 1,2,.. .
`X`	An N by P matrix with predictor variables
`S`	Positive number indicating the dimensionality of the solution
`trace`	boolean to indicate whether the user wants to see the progress of the function (default=TRUE)
`start`	either starting values (list with (U,V) or (B,V)) or way to compute them (svd, random, ca)
`maxiter`	maximum number of iterations
`dcrit`	convergence criterion

Value

Y Matrix Y from input

Xoriginal Matrix X from input

X Scaled X matrix

mx Mean values of X

sdx Standard deviations of X

ynames Variable names of responses

xnames Variable names of predictors

probabilities Estimated values of Y

m main effects

U matrix with coordinates for row-objects

B matrix with regression weight (U = XB)

V matrix with vectors for items/responses

iter number of main iterations from the MM algorithm

deviance value of the deviance at convergence

Examples

```
## Not run:
data(dataExample_clmdu)
Y<-dataExample_clmdu
X<-dataExample_clmdu
output1 = clmdu(Y)
plot(output1)
plot(output1, circles = NULL)
summary(output1)

output2 = clmdu(Y = Y, X = X)
plot(output2, circles = c(1,2))
summary(output2)

## End(Not run)
```

`clpca` *Cumulative Logistic (Restricted) PCA*

Description

Cumulative Logistic (Restricted) PCA

Usage

```
clpca(
  Y,
  X = NULL,
  S = 2,
  start = NULL,
  lambda = FALSE,
  trace = FALSE,
  maxiter = 65536,
  dcrit = 1e-06
)
```

Arguments

`Y`	An N times R ordinal matrix .
`X`	An N by P matrix with predictor variables
`S`	Positive number indicating the dimensionality of the solution
`start`	Starting values for U or B and V
`lambda`	if TRUE does lambda scaling (see Understanding Biplots, p24)
`trace`	tracing information during iterations
`maxiter`	maximum number of iterations
`dcrit`	convergence criterion

Value

Y Matrix Y from input

Xoriginal Matrix X from input

X Scaled X matrix

mx Mean values of X

sdx Standard deviations of X

ynames Variable names of responses

xnames Variable names of predictors

probabilities Estimated values of Y

m main effects

U	matrix with coordinates for row-objects
B	matrix with regression weight (U = XB)
V	matrix with vectors for items/responses
iter	number of main iterations from the MM algorithm
deviance	value of the deviance at convergence

Examples

```
## Not run:
data(dataExample_clpca)
Y<-as.matrix(dataExample_clpca[,5:8])
X<-as.matrix(dataExample_clpca[,1:4])
out = clpca(Y)
out = clpca(Y, X)

## End(Not run)
```

`dataExample_clmdu` *Dummy data for clmdu example*

Description

Dummy data for clmdu example

Usage

```
dataExample_clmdu
```

Format

A data frame with 200 observations on the following variables:

`X1`	Continuous variable 1.
`X2`	Continuous variable 2.
`X3`	Continuous variable 3.
`X4`	Continuous variable 4.
`Y1`	Discrete variable 1.
`Y2`	Discrete variable 2.
`Y3`	Discrete variable 3.
`Y4`	Discrete variable 4.
`Y5`	Discrete variable 5.

`dataExample_clpca` *Dummy data for clpca example*

Description

Dummy data for clpca example

Usage

```
dataExample_clpca
```

Format

A data frame with 200 observations on the following variables:

`X1`	Continuous variable 1.
`X2`	Continuous variable 2.
`X3`	Continuous variable 3.
`X4`	Continuous variable 4.
`Y1`	Discrete variable 1.
`Y2`	Discrete variable 2.
`Y3`	Discrete variable 3.
`Y4`	Discrete variable 4.

`dataExample_lmdu` *Dummy data for lmdu example*

Description

Dummy data for lmdu example

Usage

```
dataExample_lmdu
```

Format

A data frame with 234 observations on the following variables:

`Y1`	Dichotomous variable 1.
`Y2`	Dichotomous variable 2.
`Y3`	Dichotomous variable 3.
`Y4`	Dichotomous variable 4.

- `Y5` Dichotomous variable 5.
- `Y6` Dichotomous variable 6.
- `Y7` Dichotomous variable 7.
- `Y8` Dichotomous variable 8.
- `X1` Continuous variable 1.
- `X2` Continuous variable 2.
- `X3` Continuous variable 3.
- `X4` Continuous variable 4.
- `X5` Continuous variable 5.

`dataExample_lpca` *Dummy data for lpca example*

Description

Dummy data for lpca example

Usage

```
dataExample_lpca
```

Format

A data frame with 234 observations on the following variables:

- `Y1` Dichotomous variable 1.
- `Y2` Dichotomous variable 2.
- `Y3` Dichotomous variable 3.
- `Y4` Dichotomous variable 4.
- `Y5` Dichotomous variable 5.
- `Y6` Dichotomous variable 6.
- `Y7` Dichotomous variable 7.
- `Y8` Dichotomous variable 8.
- `X1` Continuous variable 1.
- `X2` Continuous variable 2.
- `X3` Continuous variable 3.
- `X4` Continuous variable 4.
- `X5` Continuous variable 5.

`dataExample_mru` *Dummy data for mru example*

Description

Dummy data for mru example

Usage

```
dataExample_mru
```

Format

A data frame with 234 observations on the following variables:

`y` Categorical variable.

`X1` Continuous variable 1.

`X2` Continuous variable 2.

`X3` Continuous variable 3.

`X4` Continuous variable 4.

`X5` Continuous variable 5.

`diabetes` *Diabetes data*

Description

Data description

Usage

```
data(diabetes)
```

Format

A list of 3 matrices

- X: A 145 x 3 matrix containing observed values on three predictor variables. RW = Relative weight; IR = Insulin Response; SSPG = Steady state plasma glucose.
- G: A 145 x 3 indicator matrix containing the responses on three response classes Overt, Chemical, and Non
- y: A vector of length 145 containing the responses (Overt, Chemical, Non)

References

G. M. Reaven and R. G. Miller (1979) An Attempt to Define the Nature of Chemical Diabetes Using a Multidimensional Analysis. Diabetologia 16, 17-24 D. F. Andrews A. M. Herzberg (1985). Data: A Collection of Problems from Many Fields for the Student and Research Worker. New York: Springer-Verlag Inc.

`dpes` *Dutch Parliamentary Election Study*

Description

Data description

Usage

```
data(dpes)
```

Format

A list of 3 matrices

- X: A 275 x 5 matrix containing observed values on five predictor variables. E = Euthanasia; ID = Income differences; AS = Asylum Seekers; C = Crime; LR = Left-right scaling
- G: A 275 x 8 indicator matrix containing the responses on eight response classes PvdA, CDA, VVD, D66, GL, CU, LPF, SP.
- y: A vector of length 275 containing the responses (PvdA, CDA, VVD, D66, GL, CU, LPF, SP)

References

Irwin, G., van Holsteyn, J., and den Ridder, J. (2003). Nationaal Kiezersonderzoek, NKO 2002 2003. DANS.

`fastmbu` *Fast version of mbu. It runs mbu without input checks.*

Description

Fast version of mbu. It runs mbu without input checks.

Usage

```
fastmbu(
  Y = NULL,
  W = NULL,
  XU = NULL,
  BU = NULL,
  XV = NULL,
  BV = NULL,
  mains = TRUE,
  MAXINNER = 32,
  FCRIT = 0.001,
  MAXITER = 65536,
  DCRIT = 1e-06
)
```

Arguments

`Y`	matrix with dichotomous responses
`W`	matrix with weights for each entrance of Y or vector with weights for each row of Y
`XU`	in unsupervised analysis starting values for row coordinates; in supervised analysis matrix with predictor variables for rows
`BU`	for supervised analysis matrix with regression weights for the row coordinates
`XV`	in unsupervised analysis starting values for column coordinates; in supervised analysis matrix with predictor variables for columns
`BV`	for supervised analysis matrix with regression weights for the column coordinates
`mains`	whether offsets for the items should be estimated
`MAXINNER`	maximum number of iterations in the inner loop
`FCRIT`	convergence criterion for STRESS in the inner loop
`MAXITER`	maximum number of iterations in the outer loop
`DCRIT`	convergence criterion for the deviance

Value

U estimated coordinate matrix for row objects

BU for supervised analysis the estimated matrix with regression weights for the rows

V estimated coordinate matrix for column objects

BV for supervised analysis the estimated matrix with regression weights for the columns

Mu estimated offsets

Lastinner number of iterations in the last call to STRESS

Lastfdif last difference in STRESS values in the inner loop

lastouter number of iterations in the outer loop

lastddif	last difference in deviances in outer loop
deviance	obtained deviance

`fastmru` *Fast version of mru. It runs mru without input checks.*

Description

Fast version of mru. It runs mru without input checks.

Usage

```
fastmru(
  G = NULL,
  X = NULL,
  B = NULL,
  Z = NULL,
  C = NULL,
  MAXINNER = 32,
  FCRIT = 0.001,
  MAXITER = 65536,
  DCRIT = 1e-06,
  error.check = FALSE
)
```

Arguments

`G`	indicator matrix of the response variable
`X`	matrix with predictor variables
`B`	starting values of the regression weights
`Z`	starting values for class locations
`C`	matrix with coefficients for class points, V = ZC
`MAXINNER`	maximum number of iterations in the inner loop
`FCRIT`	convergence criterion for STRESS in the inner loop
`MAXITER`	maximum number of iterations in the outer loop
`DCRIT`	convergence criterion for the deviance
`error.check`	extensive check validity input parameters (default = FALSE).

Value

B estimated regression weights

V estimated class locations

Lastinner number of iterations in the last call to STRESS

Lastfdif last difference in STRESS values in the inner loop

lastouter number of iterations in the outer loop

lastddif last difference in deviances in outer loop

deviance obtained deviance

`kieskompas` *Kieskompas data*

Description

Data description

Usage

`kieskompas`

Format

A list of 7 matrices

- G: an indicator matrix of dimension 25001 by 21 indicating the vote intention of the participants
- Xs: responses to the 30 propositions of the participants
- Xs2: responses to the 30 propositions of the participants, with NA's coded as neutral (3)
- Xb: background variables Age, Gender (male = 0, female = 1), and Education (theoretical = 0, practical = 0)
- Z: position of the 18 political parties on the 30 propositions
- stellingen: the exact Dutch wording of the propositions
- grouping: the 30 propositions can be grouped in themes (mk, bb, ef, sc, ai, bo, et, oz). mk: Environment & Climate; bb: Foreign Policy; ef: Economy & Finance; sc: Social Affairs & Culture ai: Asylum & Immigration; cc: Construction & Environment; et: Ethics; oz: Education & Healthcare

References

van Lindert, J., Meijer, S., Etienne, T., van der Steen, S., Kutiyski, Y., Moreda Laguna, O., Brousianou, A., Blanken, W., & Krouwel, A. (2023). Het Kieskompas voor de Nederlandse Tweede Kamerverkiezingen van 2023 [dataset]. Kieskompas, Amsterdam, the Netherlands.

`liver` *Liver*

Description

Data description

Usage

```
data(liver)
```

Format

A list of 3 matrices

- X: A 218 x 3 matrix containing observed values on three liver function test (predictor variables). ASpartate aminotransferase (AS), ALanine aminotransferase (AL), and Glutamate Dehydrogenase (GD)
- G: A 218 x 4 indicator matrix containing the responses on Acute Viral Hepatitis (AVH), Persistent Chronic Hepatitis (PCH), Aggressive Chronic Hepatitis (ACH), Post-Necrotic Cirrhosis (PNC).
- y: A vector of length 218 containing the responses (AVH, PCH,)

References

Plomteux, G. (1980). Multivariate analysis of an enzymic profile for the differential diagnosis of viral hepatitis. Clinical Chemistry, 26(13), 1897–1899.

`lmdu` *Logistic (Restricted) MDU*

Description

This function runs: logistic multidimensional unfolding (if X = NULL) logistic restricted multidimensional unfolding (if X != NULL)

Usage

```
lmdu(
  Y,
  f = NULL,
  X = NULL,
  S = 2,
  start = "svd",
  maxiter = 65536,
  dcrit = 1e-06
)
```

Arguments

`Y`	An N times R binary matrix .
`f`	Vector with frequencies of response patterns in Y (only applicable if (X = NULL))
`X`	An N by P matrix with predictor variables
`S`	Positive number indicating the dimensionality of the solution
`start`	Either user provided starting values (start should be a list with U and V) or a way to compute starting values (choices: random, svd, ca)
`maxiter`	maximum number of iterations
`dcrit`	convergence criterion

Value

deviance

`call`	Call to the function
`Yoriginal`	Matrix Y from input
`Y`	Matrix Y from input
`f`	frequencies of rows of Y
`Xoriginal`	Matrix X from input
`X`	Scaled X matrix
`mx`	Mean values of X
`sdx`	Standard deviations of X
`ynames`	Variable names of responses
`xnames`	Variable names of predictors
`probabilities`	Estimated values of Y
`m`	main effects
`U`	matrix with coordinates for row-objects
`B`	matrix with regression weight (U = XB)
`V`	matrix with vectors for items/responses
`iter`	number of main iterations from the MM algorithm
`deviance`	value of the deviance at convergence
`npar`	number of estimated parameters
`AIC`	Akaike's Information Criterion
`BIC`	Bayesian Information Criterion

Examples

```
## Not run:
data(dataExample_lmdu)
Y = as.matrix(dataExample_lmdu[ , 1:8])
X = as.matrix(dataExample_lmdu[ , 9:13])
# unsupervised
output = lmdu(Y = Y, S = 2)
# supervised
output2 = lmdu(Y = Y, X = X, S = 2)

## End(Not run)
```

`lpca` *Logistic (Restricted) PCA*

Description

This function runs: logistic principal component analysis (if X = NULL) logistic reduced rank regression (if X != NULL)

Usage

```
lpca(
  Y,
  X = NULL,
  S = 2,
  start = NULL,
  dim.indic = NULL,
  eq = FALSE,
  lambda = FALSE,
  maxiter = 65536,
  dcrit = 1e-06
)
```

Arguments

`Y`	An N times R binary matrix .
`X`	An N by P matrix with predictor variables
`S`	Positive number indicating the dimensionality of the solution
`start`	Option to provide starting values (list with m, U or B, and V)
`dim.indic`	An R by S matrix indicating which response variable pertains to which dimension
`eq`	Only applicable when dim.indic not NULL; equality restriction on regression weighhts per dimension
`lambda`	if TRUE does lambda scaling (see Understanding Biplots, p24)

`maxiter`	maximum number of iterations
`dcrit`	convergence criterion

Value

This function returns an object of the class `lpca` with components:

`call`	Call to the function
`Y`	Matrix Y from input
`Xoriginal`	Matrix X from input
`X`	Scaled X matrix
`mx`	Mean values of X
`sdx`	Standard deviations of X
`ynames`	Variable names of responses
`xnames`	Variable names of predictors
`probabilities`	Estimated values of Y
`m`	main effects
`U`	matrix with coordinates for row-objects
`B`	matrix with regression weight (U = XB)
`V`	matrix with vectors for items/responses
`iter`	number of main iterations from the MM algorithm
`deviance`	value of the deviance at convergence
`npar`	number of estimated parameters
`AIC`	Akaike's Information Criterion
`BIC`	Bayesian Information Criterion

Examples

```
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lpca[, 1:8])
X = as.matrix(dataExample_lpca[, 9:13])
# unsupervised
output = lpca(Y = Y, S = 2)

## End(Not run)
```

`make.df.for.varlabels` *Helper function for the plot functions*

Description

Helper function for the plot functions to add variable labels to the predictor and response variable axes

Usage

```
make.df.for.varlabels(BV, margins, names, P, R)
```

Arguments

`BV`	Concatention of matrices B (P x S) and V (R x S) or only V (R x S) from lpca
`margins`	a vector of length four indicating the margins of the plot
`names`	a vector with variable names
`P`	an integer indicating the number of predictor variables
`R`	an integer indicating the number of response variables

Value

df with information for the placement of the variable labels

`make.dfs.for.X` *Helper function for the plot functions*

Description

Helper function for the plot functions to add variable markers and labels to the predictor variable axes

Usage

```
make.dfs.for.X(Xo, P, B, xnames, mx, sdx)
```

Arguments

`Xo`	Original predictor matrix
`P`	an integer indicating the number of predictor variables
`B`	matrix (P x S) with weights
`xnames`	a vector with variable names
`mx`	averages of the original predictor matrix
`sdx`	standard deviations of the original predictor matrix

Value

output with information for the placement of variable markers and labels

---

mcd1 *Multinomial Canonical Decomposition Model for Multivariate Binary Data*

Description

The function mcd1 fits the multinomial canonical decomposition model to multivariate binary responses i.e. a double constrained reduced rank multinomial logistic model

Usage

```
mcd1(
  X,
  Y,
  S = 2,
  Z = NULL,
  W = NULL,
  ord.z = 1,
  ord.m = R,
  trace = FALSE,
  maxiter = 65536,
  dcrit = 1e-06
)
```

Arguments

`X`	An N by P matrix with predictor variables
`Y`	An N times R binary matrix .
`S`	Positive number indicating the dimensionality of teh solution
`Z`	design matrix for response
`W`	design matrix for intercepts
`ord.z`	if Z = NULL, the function creates Z having order ord.z
`ord.m`	if W = NULL, the function creates W having order ord.m
`trace`	whether progress information should be printed on the screen
`maxiter`	maximum number of iterations
`dcrit`	convergence criterion

Value

This function returns an object of the class `mcd` with components:

`call`	function call
`Xoriginal`	Matrix X from input
`X`	Scaled X matrix
`mx`	Mean values of X
`sdx`	Standard deviations of X
`Y`	Matrix Y from input
`pnames`	Variable names of profiles
`xnames`	Variable names of predictors
`znames`	Variable names of responses
`Z`	Design matrix Z
`W`	Design matrix W
`G`	Profile indicator matrix G
`m`	main effects
`bm`	regression weights for main effects
`Bx`	regression weights for X
`Bz`	regression weights for Z
`A`	regression weights (Bx Bz')
`U`	matrix with coordinates for row-objects
`V`	matrix with coordinates for column-objects
`Ghat`	Estimated values of G
`deviance`	value of the deviance at convergence
`df`	number of paramters
`AIC`	Akaike's informatoin criterion
`iter`	number of main iterations from the MM algorithm
`svd`	Singular value decomposition in last iteration

Examples

```
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lpca[ , 1:5])
X = as.matrix(dataExample_lpca[ , 9:13])
#unsupervised
output = mcd1(X, Y, S = 2, ord.z = 2)

## End(Not run)
```

`mcd2` *Multinomial Canonical Decomposition Model for a multinomial outcome*

Description

The function mcd2 fits the multinomial canonical decomposition model to a multinomial outcome i.e. a double constrained reduced rank multinomial logistic model

Usage

```
mcd2(X, G, Z, S = 2, trace = TRUE, maxiter = 65536, dcrit = 1e-06)
```

Arguments

`X`	An N by P matrix with predictor variables
`G`	An N times C class indicator matrix
`Z`	design matrix for response
`S`	Positive number indicating the dimensionality of teh solution
`trace`	whether progress information should be printed on the screen
`maxiter`	maximum number of iterations
`dcrit`	convergence criterion

Value

This function returns an object of the class `mcd` with components:

`call`	function call
`Xoriginal`	Matrix X from input
`G`	Class indicator matrix G
`X`	Scaled X matrix
`mx`	Mean values of X
`sdx`	Standard deviations of X
`pnames`	Variable names of profiles
`xnames`	Variable names of predictors
`znames`	Variable names of responses
`Z`	Design matrix Z
`m`	main effects
`Bx`	regression weights for X
`Bz`	regression weights for Z
`A`	regression weights (Bx Bz’)

`U`	matrix with coordinates for row-objects
`V`	matrix with coordinates for column-objects
`Ghat`	Estimated values of G
`deviance`	value of the deviance at convergence
`df`	number of paramters
`AIC`	Akaike's informatoin criterion
`iter`	number of main iterations from the MM algorithm
`svd`	Singular value decomposition in last iteration

Examples

```
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lpca[ , 1:5])
X = as.matrix(dataExample_lpca[ , 9:13])
#unsupervised
output = mcd1(X, Y, S = 2, ord.z = 2)

## End(Not run)
```

`mlr` *Multinomial Logistic Regression*

Description

The function mlr performs multinomial logistic regression for a nominal response variable and a set of predictor variables. It uses an MM algorithm

Usage

```
mlr(y, X, base = "largest", maxiter = 65536, dcrit = 1e-06)
```

Arguments

`y`	An N vector of the responses (categorical).
`X`	An N by P matrix with predictor variables
`base`	The category that should be used as baseline. Can be NULL, in which case the colmeans are equal to zero. Can also be "largest", in which case the
`maxiter`	maximum number of iterations
`dcrit`	convergence criterion

Value

Xoriginal	Matrix X from input
X	Scaled X matrix
G	class indicator matrix
ynames	class names of response variable
xnames	variable names of the predictors
mx	means of the predictor variables
sdx	standard deviations of the predictor variables
A	matrix with regression coefficients
iter	number of iterations
deviance	value of the deviance at convergence

Examples

```
## Not run:
data(dataExample_mru)
y = as.matrix(dataExample_mru[ , 1])
X = as.matrix(dataExample_mru[ , 2:6])
output = mlr(y = y, X = X, base = 1)

## End(Not run)
```

`mrrr` *Multinomial Reduced Rank Regression*

Description

The function mrrr performs multinomial reduced rank regression for a nominal response variable and a set of predictor variables.

Usage

```
mrrr(y, X, S = 2, trace = FALSE, maxiter = 65536, dcrit = 1e-06, start = NULL)
```

Arguments

`y`	An N vector of the responses (categorical).
`X`	An N by P matrix with predictor variables
`S`	Positive number indicating the dimensionality of teh solution
`trace`	Boolean indicating whether a trace of the algorithm should be printed on the console.
`maxiter`	maximum number of iterations
`dcrit`	convergence criterion
`start`	start values. If start=NULL, the algorithm computes the start values.

Value

Xoriginal	Matrix X from input
X	Scaled X matrix
G	class indicator matrix
ynames	class names of response classes
xnames	variable names of the predictors
mx	means of the predictor variables
sdx	standard deviations of the predictor variables
U	coordinate matrix of row objects
B	matrix with regression coefficients
V	Class coordinate matrix
iters	number of iterations
deviance	value of the deviance at convergence

Examples

```
## Not run:
data(dataExample_mru)
y = as.matrix(dataExample_mru[ , 1])
X = as.matrix(dataExample_mru[ , 2:6])
output = mrrr(y = y, X = X, S = 2)

## End(Not run)
```

`mru` *Multinomial Restricted MDU*

Description

The function mru performs multinomial restricted unfolding for a nominal response variable and a set of predictor variables.

Usage

```
mru(y, X, Z = NULL, S = 2, start = "da", maxiter = 65536, dcrit = 1e-06)
```

Arguments

`y`	An N vector of the responses (categorical) or an indicator matrix of size N x C (obligatory when Z is used)
`X`	An N by P matrix with predictor variables
`Z`	Design matrix for the class points (V)

`S`	Positive number indicating the dimensionality of the solution
`start`	Type of starting values (da: discriminant analysis, random or list with B and V)
`maxiter`	maximum number of iterations
`dcrit`	convergence criterion

Value

Y Matrix Y from input

Xoriginal Matrix X from input

X Scaled X matrix

G class indicator matrix

ynames class names of response variable

xnames variable names of the predictors

mx means of the predictor variables

sdx standard deviations of the predictor variables

U coordinate matrix of row objects

B matrix with regression coefficients

V Class coordinate matrix

iters number of iterations

deviance value of the deviance at convergence

Examples

```
## Not run:
data(dataExample_mru)
y = as.matrix(dataExample_mru[ , 1])
X = as.matrix(dataExample_mru[ , 2:6])
output = mru(y = y, X = X, S = 2)

## End(Not run)
```

`nesda` *Netherlands Study for Depression and Anxiety*

Description

Synthetic data generated on the basis of characteristics of the data described in Spinhoven e.a. (2009).

Usage

```
data(nesda)
```

Format

A list of 2 matrices

- X: A 845 x 8 matrix containing observed values on eight predictor variables. Gender (female = 1), Age, Education, Neuroticism, Extraversion, Openess, Agreeableness, Conscientiousness
- Y: A 845 x 5 matrix Indicating whether a participants suffers from a mental illness D (dysthymia), M (Major depressive disorder), G (Generalized Anxiety disorder), S (Social Phobia), P (Panic disorder)

References

Penninx BWJH, Beekman ATF, Smit JH, Zitman FG, Nolen WA, Spinhoven P, et al. The Netherlands Study of Depression and Anxiety (NESDA): rationale, objectives, and methods. International Journal of Methods in Psychiatric Research 2008;17:121–40.

Spinhoven, P., De Rooij, M., Heiser, W.J., Penninx, B. and Smit, J. (2009). The Role of Personality in Comorbidity among Anxiety and Depressive Disorders in Primary Care and Specialty Care: A Cross-Sectional Analysis. General Hospital Psychiatry, 31, 470-477.

`oos.comparison` *This function compares the predictive performance of several models fitted on the same data*

Description

The number of bootstraps should be the same for each model Ideally, the seed used in bootstrapping should also be the same

Usage

```
oos.comparison(objectlist, xlabel = "Model")
```

Arguments

`objectlist`	An list with output objects from the bootstrap functions in lmap
`xlabel`	A character object, specifying the label on the horizontal axis. Default is "Model"

Value

plot A boxplot with prediction errors for each model

pe A data frame with average prediction error for each bootstrap

fit A matrix with prediction error statistics for each model

Examples

```
## Not run:
data(dataExample_mru)
y = as.matrix(dataExample_mru[ , 1])
X = as.matrix(dataExample_mru[ , 2:6])
output2 = mrrr(y = y, X = X, S = 2)
b2 = bootstrap.mrrr(output2)
output3 = mrrr(y = y, X = X, S = 3)
b3 = bootstrap.mrrr(output3)
myobjects = list(b2, b3)
comparison = oos.comparison(myobjects)
comparison$plot
comparison$fit

## End(Not run)
```

`plot.bootstrap` *Plot an object obtained using one of the bootstrap functions*

Description

Plot an object obtained using one of the bootstrap functions

Arguments

`x`	an object of type bootstrap
`level`	level of confidence regios
`type`	choose between "Bag" (default) or "norm"
`...`	additional arguments to be passed.
`prop`	Proportion of all the points to be included in the bag (default is 0.5)

Value

Plot of the results obtained from bootstrap

`plot.clmdu` *Plots a Cumulative Logistic MDU model*

Description

Plots a Cumulative Logistic MDU model

Usage

```
## S3 method for class 'clmdu'
plot(
  x,
  dims = c(1, 2),
  circles = seq(1, R),
  ycol = "darkgreen",
  xcol = "darkblue",
  ocol = "grey",
  markersize = 2.5,
  labelsize = 3,
  ...
)
```

Arguments

`x`	an object of type clmdu
`dims`	which dimensions to visualize
`circles`	which circles to visualize
`ycol`	colour for representation of response variables
`xcol`	colour for representation of predictor variables
`ocol`	colour for representation of row objects
`markersize`	size of points
`labelsize`	size of labels
`...`	additional arguments to be passed.

Value

Plot of the results obtained from clmdu

Examples

```
## Not run:
data(dataExample_clmdu)
Y = as.matrix(dataExample_clmdu[ , 1:8])
X = as.matrix(dataExample_clmdu[ , 9:13])
# unsupervised
output = clmdu(Y = Y, S = 2)
plot(output)

## End(Not run)
```

`plot.clpca` *Plots a Cumulative Logistic PCA model*

Description

Plots a Cumulative Logistic PCA model

Usage

```
## S3 method for class 'clpca'
plot(
  x,
  dims = c(1, 2),
  ycol = "darkgreen",
  xcol = "darkblue",
  ocol = "grey",
  markersize = 2.5,
  labelsize = 3,
  ...
)
```

Arguments

`x`	an object of type clpca
`dims`	which dimensions to visualize
`ycol`	colour for representation of response variables
`xcol`	colour for representation of predictor variables
`ocol`	colour for representation of row objects
`markersize`	size of points
`labelsize`	size of labels
`...`	additional arguments to be passed.

Value

Plot of the results obtained from clpca

Examples

```
## Not run:
data(dataExample_clpca)
Y<-as.matrix(dataExample_clpca[,5:8])
X<-as.matrix(dataExample_clpca[,1:4])
out = clpca(Y, X)
plot(out)

## End(Not run)
```

`plot.lmdu` *Plots a Logistic MDU model*

Description

Plots a Logistic MDU model

Usage

```
## S3 method for class 'lmdu'
plot(
  x,
  dims = c(1, 2),
  ycol = "darkgreen",
  xcol = "darkblue",
  ocol = "grey",
  markersize = 2.5,
  labelsize = 3,
  ...
)
```

Arguments

`x`	an object of type lmdu
`dims`	which dimensions to visualize
`ycol`	colour for representation of response variables
`xcol`	colour for representation of predictor variables
`ocol`	colour for representation of row objects
`markersize`	size of points
`labelsize`	size of labels
`...`	additional arguments to be passed.

Value

Plot of the results obtained from lmdu

Examples

```
## Not run:
data(dataExample_lmdu)
Y = as.matrix(dataExample_lmdu[ , 1:8])
X = as.matrix(dataExample_lmdu[ , 9:13])
# unsupervised
output = lmdu(Y = Y, S = 2)
plot(output)
```

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

`plot.lpca` *Plots a Logistic PCA Model*

Description

Plots a Logistic PCA Model

Usage

```
## S3 method for class 'lpca'
plot(
  x,
  dims = c(1, 2),
  type = "H",
  pmarkers = seq(0.1, 0.9, by = 0.1),
  ycol = "darkgreen",
  xcol = "darkblue",
  ocol = "grey",
  ...
)
```

Arguments

`x`	an object of type lpca
`dims`	which dimensions to visualize
`type`	either H (hybrid), I (inner product/pca), or D (distance/melodic)
`pmarkers`	vector or list of length R (the number of response variables) with values between 0 and 1 for markers on the response variable axes.
`ycol`	colour for representation of response variables
`xcol`	colour for representation of predictor variables
`ocol`	colour for representation of row objects
`...`	additional arguments to be passed.

Value

Plot of the results obtained from lpca

Examples

```
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lpca[, 1:8])
X = as.matrix(dataExample_lpca[, 9:13])
# unsupervised
output = lpca(Y = Y, S = 2)
plot(output)

## End(Not run)
```

`plot.mrrr` *Plots a Multinomial Reduced Rank Model*

Description

Plots a Multinomial Reduced Rank Model

Usage

```
## S3 method for class 'mrrr'
plot(
  x,
  dims = c(1, 2),
  ycol = "darkgreen",
  xcol = "darkblue",
  ocol = "grey",
  markersize = 2.5,
  labelsize = 3,
  ...
)
```

Arguments

`x`	an object of type mrrr
`dims`	which dimensions to visualize
`ycol`	colour for representation of response variables
`xcol`	colour for representation of predictor variables
`ocol`	colour for representation of row objects
`markersize`	determines the size of the markera
`labelsize`	determines the size of the labels
`...`	additional arguments to be passed.

Value

Plot of the results obtained from mrrr

Examples

```
## Not run:
data(dataExample_mru)
y = as.matrix(dataExample_mru[ , 1])
X = as.matrix(dataExample_mru[ , 2:6])
output = mru(y = y, X = X, S = 2)
plot(output)

## End(Not run)
```

`plot.mru` *Plots a Multinomial Restricted MDU model*

Description

Plots a Multinomial Restricted MDU model

Usage

```
## S3 method for class 'mru'
plot(
  x,
  dims = c(1, 2),
  class.regions = FALSE,
  ycol = "darkgreen",
  xcol = "darkblue",
  ocol = "grey",
  markersize = 2.5,
  labelsize = 3,
  ...
)
```

Arguments

`x`	an object of type mru
`dims`	which dimensions to visualize
`class.regions`	whether a voronoi diagram with classification regions should be included
`ycol`	colour for representation of response variables
`xcol`	colour for representation of predictor variables
`ocol`	colour for representation of row objects
`markersize`	size of points
`labelsize`	size of labels
`...`	additional arguments to be passed.

Value

Plot of the results obtained from mru

Examples

```
## Not run:
data(dataExample_mru)
y = as.matrix(dataExample_mru[ , 1])
X = as.matrix(dataExample_mru[ , 2:6])
output = mru(y = y, X = X, S = 2)
plot(output)

## End(Not run)
```

`plot.trioscale` *Plotting function for object of class trioscale*

Description

Plotting function for object of class trioscale

Usage

```
## S3 method for class 'trioscale'
plot(
  x,
  ycol = "darkgreen",
  xcol = "darkblue",
  ocol = "grey",
  markersize = 2.5,
  labelsize = 3,
  classlabels = NULL,
  s1 = 2.5,
  s2 = 1.05,
  s3 = 1.15,
  ...
)
```

Arguments

`x`	An object of class trioscale.
`ycol`	colour for representation of response variables
`xcol`	colour for representation of predictor variables
`ocol`	colour for representation of row objects
`markersize`	size of points

`labelsize`	size of labels
`classlabels`	List with plotting options for the labels of the Anchor points
`s1`	scaling factor for distance between points and log-ratio axes
`s2`	scaling factor for positioning class labels
`s3`	scaling factor for positioning variable lables
`...`	additional arguments to be passed.

Value

This function returns an plot

Examples

```
## Not run:
out = trioscale(data)
plot.trioscale(out)

## End(Not run)
```

`predict.clmdu` *The function predict.clmdu makes predictions for a test/validation set based on a fitted cl restricted multidimensional unfolding model (clmdu with X)*

Description

The function predict.clmdu makes predictions for a test/validation set based on a fitted cl restricted multidimensional unfolding model (clmdu with X)

Usage

```
## S3 method for class 'clmdu'
predict(object, newX, newY = NULL, ...)
```

Arguments

`object`	An `clmdu` object
`newX`	An N by P matrix with predictor variables for a test/validation set
`newY`	An N by R matrix with response variables for a test/validation set
`...`	additional arguments to be passed.

Value

This function returns an object of the class `predclpca` with components:

`Yhat`	Predicted values for the test set
`devr`	Estimated prediction deviance for separate responses
`devtot`	Estimated prediction deviance for all responses

Examples

```
## Not run:
data(dataExample_clpca)
Y = as.matrix(dataExample_clmdu[ , 1:8])
X = as.matrix(dataExample_clmdu[ , 9:13])
newY = as.matrix(dataExample_clmdu[1:20 , 1:8])
newX = as.matrix(dataExample_clmdu[1:20 , 9:13])
# supervised
output = clmdu(Y = Y, X = X, S = 2)
preds = predict(output, newX = newX, newY = newY)

## End(Not run)
```

`predict.clpca` *The function predict.clpca makes predictions for a test/validation set based on a fitted clrrr model (clpca with X)*

Description

The function predict.clpca makes predictions for a test/validation set based on a fitted clrrr model (clpca with X)

Usage

```
## S3 method for class 'clpca'
predict(object, newX, newY = NULL, ...)
```

Arguments

`object`	An `clpca` object
`newX`	An N by P matrix with predictor variables for a test/validation set
`newY`	An N by R matrix with response variables for a test/validation set
`...`	additional arguments to be passed.

Value

This function returns an object of the class `predclpca` with components:

`Yhat`	Predicted values for the test set
`devr`	Estimated prediction deviance for separate responses
`devtot`	Estimated prediction deviance for all responses

Examples

```
## Not run:
data(dataExample_clpca)
Y = as.matrix(dataExample_clpca[ , 1:8])
X = as.matrix(dataExample_clpca[ , 9:13])
newY = as.matrix(dataExample_clpca[1:20 , 1:8])
newX = as.matrix(dataExample_clpca[1:20 , 9:13])
# supervised
output = clpca(Y = Y, X = X, S = 2)
preds = predict(output, newX = newX, newY = newY)

## End(Not run)
```

`predict.lmdu` *The function predict.lmdu makes predictions for a test/validation set based on a fitted lrmdu model (lmdu with X)*

Description

The function predict.lmdu makes predictions for a test/validation set based on a fitted lrmdu model (lmdu with X)

Usage

```
## S3 method for class 'lmdu'
predict(object, newX, newY = NULL, ...)
```

Arguments

`object`	An `lmdu` object
`newX`	An N by P matrix with predictor variables for a test/validation set
`newY`	An N by R matrix with response variables for a test/validation set
`...`	additional arguments to be passed.

Value

This function returns an object of the class `lpca` with components:

`Yhat`	Predicted values for the test set
`devr`	Estimated prediction deviance for separate responses
`devtot`	Estimated prediction deviance for all responses
`Brier.r`	Estimated Brier score for separate responses
`Brier`	Estimated Brier score for all responses

Examples

```
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lmdu[-c(1:20) , 1:8])
X = as.matrix(dataExample_lmdu[-c(1:20) , 9:13])
newY = as.matrix(dataExample_lmdu[1:20 , 1:8])
newX = as.matrix(dataExample_lmdu[1:20 , 9:13])
# supervised
output = lmdu(Y = Y, X = X, S = 2)
preds = predict(output, newX = newX, newY = newY)

## End(Not run)
```

`predict.lpca` *The function predict.lpca makes predictions for a test/validation set based on a fitted lrrr model (lpca with X)*

Description

The function predict.lpca makes predictions for a test/validation set based on a fitted lrrr model (lpca with X)

Usage

```
## S3 method for class 'lpca'
predict(object, newX, newY = NULL, ...)
```

Arguments

`object`	An `lpca` object
`newX`	An N by P matrix with predictor variables for a test/validation set
`newY`	An N by R matrix with response variables for a test/validation set
`...`	additional arguments to be passed.

Value

This function returns an object of the class `lpca` with components:

`theta`	Predicted canonical values for the test set
`Yhat`	Predicted values for the test set
`devr`	Estimated prediction deviance for separate responses
`devtot`	Estimated prediction deviance for all responses
`Brier.r`	Estimated Brier score for separate responses
`Brier`	Estimated Brier score for all responses

Examples

```
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lpca[-c(1:20) , 1:8])
X = as.matrix(dataExample_lpca[-c(1:20) , 9:13])
newY = as.matrix(dataExample_lpca[1:20 , 1:8])
newX = as.matrix(dataExample_lpca[1:20 , 9:13])
# supervised
output = lpca(Y = Y, X = X, S = 2)
preds = predict(output, newX = newX, newY = newY)

## End(Not run)
```

`predict.mlr` *The function predict.mlr makes predictions for a test/validation set based on a fitted mlr model*

Description

The function predict.mlr makes predictions for a test/validation set based on a fitted mlr model

Usage

```
## S3 method for class 'mlr'
predict(object, newX, ...)
```

Arguments

`object`	An `mlr` object
`newX`	An N by P matrix with predictor variables for a test/validation set
`...`	additional arguments to be passed.

Value

This function returns an object of the class `p.mlr` with components:

`Ghat`	Predicted values (probabilities) for the test set

Examples

```
## Not run:
data(dataExample_mru)
y = as.matrix(dataExample_mru[ , 1])
X = as.matrix(dataExample_mru[ , 2:6])
output = mlr(y = y, X = X, base = 1)
preds = predict(output, newX = X[1:4, ])

## End(Not run)
```

`predict.mrrr` *The function predict.mrrr makes predictions for a test/validation set based on a fitted mrrr model*

Description

The function predict.mrrr makes predictions for a test/validation set based on a fitted mrrr model

Usage

```
## S3 method for class 'mrrr'
predict(object, newX, ...)
```

Arguments

`object`	An `mrrr` object
`newX`	An N by P matrix with predictor variables for a test/validation set
`...`	additional arguments to be passed.

Value

This function returns an object of the class `p.mru` with components:

`Ghat`	Predicted values for the test set

Examples

```
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_mru[-c(1:20) , 1:8])
X = as.matrix(dataExample_mru[-c(1:20) , 9:13])
newY = as.matrix(dataExample_mru[1:20 , 1:8])
newX = as.matrix(dataExample_mru[1:20 , 9:13])
output = mrrr(Y = Y, X = X, S = 2)
preds = predict(output, newX = newX)

## End(Not run)
```

`predict.mru` *The function predict.mru makes predictions for a test/validation set based on a fitted mru model*

Description

The function predict.mru makes predictions for a test/validation set based on a fitted mru model

Usage

```
## S3 method for class 'mru'
predict(object, newX, newG = NULL, ...)
```

Arguments

`object`	An `mru` object
`newX`	An N by P matrix with predictor variables for a test/validation set
`newG`	An N by R matrix with response variables for a test/validation set
`...`	additional arguments to be passed.

Value

This function returns an object of the class `p.mru` with components:

`Yhat`	Predicted values for the test set
`dev`	Estimated prediction deviance

Examples

```
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_mru[-c(1:20) , 1:8])
X = as.matrix(dataExample_mru[-c(1:20) , 9:13])
newY = as.matrix(dataExample_mru[1:20 , 1:8])
newX = as.matrix(dataExample_mru[1:20 , 9:13])
# supervised
output = mru(Y = Y, X = X, S = 2)
preds = predict(output, newX = newX, newY = newY)

## End(Not run)
```

`procrustes1` *Two procedures for procrustes analysis*

Description

procrustes1 does orthogonal procrustes analysis procrustes2 does similarity procrustes analysis

Usage

```
procrustes1(V1, V2)
```

Arguments

`V1`	first coordinate matrix
`V2`	second coordinate matrix

Value

either a rotation matrix (procrustes1) or a list with a rotation matrix (R), a stretching factor (s) and a translation (t) (procrustes2)

`procx` *Helper function for pre-processing the predictors*

Description

Continuous predictor variables are standardized - mean zero, standard deviation one. Categorical predictor variables are represented as dummy (0, 1) variables.

Usage

```
procx(X)
```

Arguments

`X`	An N by P matrix with predictor variables

Value

Xoriginal

dichotomous indicator which predictor variables are dichotomous

X standardized matrix

mx averages of original variables

sdx standard deviation of original variables

`read_drugdata` *Function for reading the drug consumption data from the UCI repository*

Description

Function for reading the drug consumption data from the UCI repository

Usage

```
read_drugdata()
```

Value

X first coordinate matrix

Y matrix with response variables (Alcohol,Am,Amyl,Be,Caff,Ca,Choc,Co,Crack,Ex,Heroin,Ke,Le,LSD,Me,Mu,Ni,Semer,V

idx indicator which response variables have a probability between 0.1 and 0.9

References

Fehrman, E., Muhammad, A. K., Mirkes, E. M., Egan, V., & Gorban, A. N. (2017). The five factor model of personality and evaluation of drug consumption risk. In Data science: innovative developments in data analysis and clustering (pp. 231-242). Springer International Publishing.

`read_isspdata_peb` *Function to read in the ISSP data It requires the file ZA7650_v1-0-0.sav to be on your computer this file can be obtained from /www.gesis.org/en/issp/modules/issp-modules-by-topic/environment/2020 ZA7650 Data file Version 1.0.0, https://doi.org/10.4232/1.13921.*

Description

Function to read in the ISSP data It requires the file ZA7650_v1-0-0.sav to be on your computer this file can be obtained from /www.gesis.org/en/issp/modules/issp-modules-by-topic/environment/2020 ZA7650 Data file Version 1.0.0, https://doi.org/10.4232/1.13921.

Usage

```
read_isspdata_peb(path)
```

Arguments

`path`	path where the ZA7650_v1-0-0.sav file is saved

Value

X matrix containing the predictor variables

Y matrix with response variables

References

ISSP Research Group (2022). International social survey programme: Environment IV - issp 2020. GESIS, Cologne.

`summary.clmdu` *Summarizing Cumulative Logistic MDU models The function summary.lmdu gives a summary from an object from clmdu()*

Description

Summarizing Cumulative Logistic MDU models

The function summary.lmdu gives a summary from an object from clmdu()

Usage

```
## S3 method for class 'clmdu'
summary(object, ...)
```

Arguments

`object`	An object resulting from clmdu
`...`	additional arguments to be passed.

Value

Summary of the results obtained from clmdu

`summary.clpca` *Summarizing Cumulative Logistic PCA models*

Description

The function summary.clpca gives a summary from an object from clpca()

Usage

```
## S3 method for class 'clpca'
summary(object, ...)
```

Arguments

`object`	An object resulting from clpca
`...`	additional arguments to be passed.

Value

Summary of the results obtained from clpca

`summary.lmdu` *Summarizing Logistic MDU models*

Description

The function summary.lmdu gives a summary from an object from lmdu()

Usage

```
## S3 method for class 'lmdu'
summary(object, ...)
```

Arguments

`object`	An object resulting from lmdu
`...`	additional arguments to be passed.

Value

Summary of the results obtained from lmdu

`summary.lpca` *Summarizing Logistic PCA models*

Description

The function summary.lpca gives a summary from an object from lpca()

Usage

```
## S3 method for class 'lpca'
summary(object, ...)
```

Arguments

`object`	An object resulting from lpca
`...`	additional arguments to be passed.

Value

Summary of the results obtained from lpca

`summary.mcd` *Summarizing an Multinomial Canonical Decomposition Model*

Description

The function summary.mcd1 gives a summary from an object from mcd()

Usage

```
## S3 method for class 'mcd'
summary(object, ...)
```

Arguments

`object`	An object resulting from mcd
`...`	additional arguments to be passed.

Value

Summary of the results obtained from mcd

`summary.mlr` *Summarizing Multinomial Logistic Regression Model*

Description

The function summary.mlr gives a summary from an object from mlr()

Usage

```
## S3 method for class 'mlr'
summary(object, ...)
```

Arguments

`object`	An object resulting from mlr
`...`	additional arguments to be passed.

Value

Summary of the results obtained from mlr

`summary.mrrr` *Summarizing Multinomial Reduced Rank Model*

Description

The function summary.mrrr gives a summary from an object from mrrr()

Usage

```
## S3 method for class 'mrrr'
summary(object, ...)
```

Arguments

`object`	An object resulting from mrrr
`...`	additional arguments to be passed.

Value

Summary of the results obtained from mrrr

`summary.mru` *Summarizing Multinomial Restricted Unfolding Model The function summary.mru gives a summary from an object from mru()*

Description

Summarizing Multinomial Restricted Unfolding Model

The function summary.mru gives a summary from an object from mru()

Usage

```
## S3 method for class 'mru'
summary(object, ...)
```

Arguments

`object`	An object resulting from mru
`...`	additional arguments to be passed.

Value

Summary of the results obtained from mru

`summary.trioscale` *Summarizing TrioScale*

Description

The function summary.trioscale gives a summary from the MLR in trioscale

Usage

```
## S3 method for class 'trioscale'
summary(object, ...)
```

Arguments

`object`	An object resulting from trioscale
`...`	additional arguments to be passed.

Value

Summary of the results obtained from trioscale

`theme_lmda` *Theme_lmda*

Description

produces the logistic MDA theme for ggplots

Usage

```
theme_lmda()
```

`trioscale` *Function for TRIOSCALE*

Description

Function for TRIOSCALE

Usage

```
trioscale(y, X)
```

Arguments

`y`	A response formula with 3 classes
`X`	A predictor matrix

Value

This function returns an object of class trioscale

`data`	data
`mlr`	Output object from ts.mlr
`Q`	result from P2Q
`X`	X matrix with coordinates
`Xdf`	X as a data frame

Examples

```
## Not run:
data(diabetes)
output = trioscale(y = diabetes$y, X = diabetes$X)
plot(output)

## End(Not run)
```

`twomodedistance` *The function twomodedistance computes the two mode (unfolding) distance*

Description

The function twomodedistance computes the two mode (unfolding) distance

Usage

```
twomodedistance(U, V)
```

Arguments

`U`	An N times S matrix with coordinates in S dimensional Euclidean space.
`V`	An R times S matrix with coordinates in S dimensional Euclidean space.

Value

D a N by R matrix with Euclidean distances

Index

∗ **datasets**
    dataExample_clmdu, 11
    dataExample_clpca, 12
    dataExample_lmdu, 12
    dataExample_lpca, 13
    dataExample_mru, 14
∗ **dataset**
    diabetes, 14
    dpes, 15
    kieskompas, 18
    liver, 19
    nesda, 30

bootstrap.clmdu, 3
bootstrap.clpca, 4
bootstrap.lmdu, 4
bootstrap.lpca, 5
bootstrap.mcd, 6
bootstrap.mrrr, 7
bootstrap.mru, 7

clmdu, 8
clpca, 10

dataExample_clmdu, 11
dataExample_clpca, 12
dataExample_lmdu, 12
dataExample_lpca, 13
dataExample_mru, 14
diabetes, 14
dpes, 15

fastmbu, 15
fastmru, 17

kieskompas, 18

liver, 19
lmdu, 19
lpca, 21

make.df.for.varlabels, 23
make.dfs.for.X, 23
mcd1, 24
mcd2, 26
mlr, 27
mrrr, 28
mru, 29

nesda, 30

oos.comparison, 31

plot.bootstrap, 32
plot.clmdu, 32
plot.clpca, 34
plot.lmdu, 35
plot.lpca, 36
plot.mrrr, 37
plot.mru, 38
plot.trioscale, 39
predict.clmdu, 40
predict.clpca, 41
predict.lmdu, 42
predict.lpca, 43
predict.mlr, 44
predict.mrrr, 45
predict.mru, 46
procrustes1, 47
procx, 47

read_drugdata, 48
read_isspdata_peb, 49

summary.clmdu, 49
summary.clpca, 50
summary.lmdu, 50
summary.lpca, 51
summary.mcd, 51
summary.mlr, 52
summary.mrrr, 52
summary.mru, 53

summary.trioscale, 53

theme_lmda, 54
trioscale, 54
twomodedistance, 55