

# Package ‘smacof’

August 10, 2015

**Type** Package

**Title** Multidimensional Scaling

**Version** 1.7-0

**Date** 2015-08-06

**Author** Patrick Mair [aut, cre], Jan De Leeuw [aut], Ingwer Borg [ctb], Patrick J. F. Groenen [aut]

**Maintainer** Patrick Mair <mair@fas.harvard.edu>

**Description** Provides the following approaches for multidimensional scaling (MDS) based on stress minimization by means of majorization (smacof): basic MDS on symmetric dissimilarity matrices, MDS with external constraints on the configuration, individual difference scaling (idioscal, ind-scal, and friends), MDS with spherical restrictions, and unfolding. The MDS type can be ratio, interval, ordinal, and monotone splines. Various tools and extensions like jackknife MDS, permutation tests, gravity models, inverse MDS, unidimensional scaling, classical scaling, and Procrustes.

**Imports** graphics, stats, polynom, Hmisc, colorspace, nnls, grDevices

**Depends** R (>= 3.0.2)

**License** GPL-3

**Suggests** knitr, RgoogleMaps

**VignetteBuilder** knitr

**LazyData** yes

**LazyLoad** yes

**ByteCompile** yes

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2015-08-10 09:21:42

**R topics documented:**

bread	3
breakfast	3
csrranking	4
Duration	5
ekman	6
EW_ger	6
GOPdtm	7
gravity	8
Guerry	9
helm	9
intelligence	10
inverseMDS	11
jackknife	12
kinshipdelta	14
LawLer	15
morse	15
morsescales	16
OCP	17
partypref	17
perception	18
permtest	19
Plato7	20
plot.smacof	21
Procrustes	24
PVQ40	26
randomstress	27
rectangles	28
residuals.smacof	28
RockHard	29
sim2diss	30
smacofConstraint	31
smacofIndDiff	34
smacofRect	36
smacofSphere	38
smacofSym	40
stardist	42
summary.smacofB	43
torgerson	44
trading	44
uniscale	46
winedat	47
wish_ger	47

---

bread	<i>Breakfast preferences</i>
-------	------------------------------

---

**Description**

The data set is described in Bro (1998). The raw data consist of ratings of 10 breads on 11 different attributes carried out by 8 raters. Note that the bread samples are pairwise replications: Each of the 5 different breads, which have a different salt content, was presented twice for rating.

**Usage**

```
data(bread)
```

**Format**

A list of length 8 with elements of class "dist". The attributes are bread odor, yeast odor, off-flavor, color, moisture, dough, salt taste, sweet taste, yeast taste, other taste, and total taste.

**References**

Bro, R. (1998). Multi-way Analysis in the Food Industry: Models, Algorithms, and Applications. Ph.D. thesis, University of Amsterdam (NL) & Royal Veterinary and Agricultural University (DK).

**Examples**

```
data(bread)
```

---

breakfast	<i>Breakfast preferences</i>
-----------	------------------------------

---

**Description**

42 individuals were asked to order 15 breakfast items due to their preference.

**Usage**

```
data(breakfast)
```

**Format**

Data frame with students in the rows and breakfast items in the columns.

toast: toast pop-up

butoast: buttered toast

engmuff: English muffin and margarine

jdonut: jelly donut

cintoast: cinnamon toast

bluemuff: blueberry muffin and margarine

hrolls: hard rolls and butter

toastmarm: toast and marmalade

butoastj: buttered toast and jelly

toastmarg: toast and margarine

cinbun: cinnamon bun

danpastry: Danish pastry

gdonut: glazed donut

cofcake: coffee cake

cornmuff: corn muffin and butter

**References**

Green, P. E. & Rao, V. (1972). Applied multidimensional scaling. Hinsdale, IL: Dryden.

**Examples**

```
data(breakfast)
```

---

csrranking

*CSR activities*

---

**Description**

This dataset collects rankings of 100 individual on 5 topics that reflect social responsibilities on corporations.

**Usage**

```
data(csrranking)
```

**Format**

A data frame where each individual ranked prevention of environmental pollution (Environment), waste prevention (Waste Prevention), selling organic products (Organic Products), participating on charity programs (Charity), and fair treatment of employees (Employee) according to its own preferences. A value of 1 corresponds to highest importance, 5 to lowest importance.

**Examples**

```
data(csrranking)
```

---

Duration

*Facets of the subjective duration of imagined situations*

---

**Description**

The DurationRaw dataset contains the duration rating of 76 subjects on 24 situations. Subjects were asked to rate the duration on a 7 point scale (1 ... substantially shorter, 7 ... substantially longer). The Duration data file contains the corresponding correlations between the 24 situations including some information about the facets.

**Usage**

```
data(Duration)  
data(DurationRaw)
```

**Format**

Data frame 24 correlations based on duration ratings:

S1-S24: situation

F1: pleasant (1), neutral (2), unpleasant (3)

F2: variable (1), monotonous (2)

F3: difficult (1), easy (2)

F3: many (1), few (2)

structuple: the facet structure written as a tuple

**References**

Borg, I., & Groenen, P. J. F. (2005). Modern Multidimensional Scaling (2nd ed.). Springer.

**Examples**

```
ddiss <- sim2diss(Duration[,paste0("S", 1:24)])  
fit <- mds(ddiss, type = "ordinal", ndim = 4)  
plot(fit)
```

ekman

*Ekman data set*

---

**Description**

Ekman dissimilarities

**Usage**`data(ekman)`**Format**Object of class `dist`**Details**

Ekman presents similarities for 14 colors which are based on a rating by 31 subjects where each pair of colors was rated on a 5-point scale (0 = no similarity up to 4 = identical). After averaging, the similarities were divided by 4 such that they are within the unit interval. Similarities of colors with wavelengths from 434 to 674 nm.

**References**

Ekman, G. (1954). Dimensions of color vision. *Journal of Psychology*, 38, 467-474.

**Examples**

```
data(ekman)
## maybe str(ekman) ; plot(ekman) ...
```

---

EW\_ger*Work values*

---

**Description**

Intercorrelations of 13 working values for former West (first list element) and East Germany.

**Usage**`data(EW_eng)`**Format**Object of class `dist`

**Details**

Note that in `EW_ger` the labels are given in German. For `smacof`, the data must be converted into a dissimilarity matrix by applying the `sim2diss()` function to each list element.

**References**

ALLBUS 1991, German General Social Survey.

Borg, I., Groenen, P. J. F., & Mair, P. (2010). *Multidimensionale Skalierung*. Muenchen: Hampp Verlag.

Borg, I., Groenen, P. J. F., & Mair, P. (2012). *Multidimensional Scaling*. New York: Springer, forthcoming.

**Examples**

```
data(EW_eng)
data(EW_ger)
```

---

GOPdtm

*Republican Statements*

---

**Description**

Document-term matrix based on statements by Republican voters.

**Usage**

```
data(GOPdtm)
```

**Format**

Document-term matrix with statements in the rows and terms (keywords) in the columns

**Details**

This dataset emerges from statements of Republican voters scraped from the official GOP website. They were asked to complete the sentence "I am a Republican because ...". We have selected the 37 most frequent words and created a document-term matrix.

**References**

Mair, P., Rusch, T. & Hornik, K. (2014). Gravity MDS on sparse document-term matrices: Republican values. Research Report Series, Department of Statistics and Mathematics, WU Vienna, Report 127.

**Examples**

```
data(GOPdtm)
GOPdtm
```

gravity

*Gravity dissimilarities*

---

**Description**

Computes the dissimilarities using a gravity model based on co-occurrences

**Usage**

```
gravity(X, lambda = 1)
```

**Arguments**

X	numeric matrix
lambda	tuning parameter

**Details**

The first step in this function is to compute the co-occurrences. Based on the binarized data matrix  $Y$  we compute  $Y'Y$  which leads to the co-occurrence matrix. We then use the gravity model to compute the gravity dissimilarities. In order to give more (or less) structure to the MDS solution, the tuning parameter (which defines a power transformation) can be increased (or decreased with lower bound 0). In addition, a weight matrix is created that sets cells with no co-occurrences to 0, i.e. they are blanked out in a subsequent smacof fit (in the gravdiss output they are fixed to a value of 1000).

**Value**

gravdiss	Gravity dissimilarities
weightmat	Weight matrix for subsequent smacof computation
co.occ	Matrix with co-occurrences

**Author(s)**

Patrick Mair

**References**

Mair, P., Rusch, T. & Hornik, K. (2014). Gravity MDS on sparse document-term matrices: Republican values. Research Report Series, Department of Statistics and Mathematics, WU Vienna, Report 127.

**See Also**

[smacofSym](#)



**Examples**

```
data(GOPdtm)
gravD <- gravity(GOPdtm, lambda = 2)
res <- smacofSym(gravD$gravdiss, weightmat = gravD$weightmat)
plot(res)
```

---

Guerry

*Map Dataset France 1830*

---

**Description**

Airline distances (in km) between French Departments in 1830.

**Usage**

```
data(Guerry)
```

**Format**

Symmetric matrix with distances.

**References**

Friendly, M. (2007). A. M. Guerry's Moral Statistics of France: Challenges for Multivariate Spatial Analysis, *Statistical Science*, 2007, 22(3), 368-399.

**Examples**

```
data(Guerry)
```

---

helm

*Helm's color data*

---

**Description**

This dataset contains dissimilarity data for individual difference scaling from an experiment carried out by Helm (1959).

**Usage**

```
data(helm)
```

**Format**

List containing objects of class `dist`

**Details**

A detailed description of the experiment can be found in Borg and Groenen (2005, p. 451) with the corresponding Table 21.1. containing distance estimates for color pairs. There were 14 subjects that rated the similarity of colors, 2 of whom replicated the experiment. 10 subjects have a normal color vision (labelled by N1 to N10 in our list object), 4 of them are red-green deficient in varying degrees. In this dataset we give the dissimilarity matrices for each of the subjects, including the replications. They are organized as a list of length 16 suited for smacofIndDiff computations.

The authors thank Michael Friendly and Phil Spector for data preparation.

**References**

Helm, C. E. (1959). A multidimensional ratio scaling analysis of color relations. Technical Report, Princeton University and Educational Testing Service. Princeton, NJ.

Borg, I., & Groenen, P. J. F. (2005). Modern Multidimensional Scaling: Theory and Applications (2nd edition). New York: Springer.

**Examples**

```
data(helm)
length(helm)
helm
```

---

intelligence

*Intelligence Tests*

---

**Description**

Contains intercorrelations of 8 intelligence tests, together with two facets. In addition, a hypothesized restriction matrix for the intercorrelations is provided. The proximities for items with the same structuples, such as p(NA1,NA2) and p(GA1,GA3), all are set to the value 5. The proximities that correspond to the immediate neighborhood relations are set to the value 4, since none of these distances should be larger than any distance between definitionally equivalent items. Finally, the large distances between the groups NI, GA and the groups NA, GI are set to 3. The intelligence tests are coded on the following facets: format (N = numerical, G = geometrical) and requirement (A = application, I - inference).

**Usage**

```
data(intelligence)
```

**Format**

Data frame of 8 intelligence tests: facets, intercorrelations, and restrictions

Test: Test number

Language: numerical, geometrical

Requirement: application, inference

T1-T8: intercorrelations

R1-R8: restrictions

## References

Borg, I., & Groenen, P. J. F. (2005). *Modern Multidimensional Scaling* (2nd ed.). Springer.

## Examples

```
idiss <- sim2diss(intelligence[,paste0("T", 1:8)])
fit <- mds(idiss)
plot(fit)
```

---

inverseMDS

*Inverse MDS*

---

## Description

Performs inverse MDS. That is, given a configuration matrix, it computes a dissimilarity matrix.

## Usage

```
inverseMDS(x)
```

## Arguments

x                    Configuration matrix

## Details

Note that this is a very basic implementation. It can be that some dissimilarities are negative. Also, the weights are fixed to a value of 1.

## Value

Returns a list with dissimilarity matrices.

## References

De Leeuw J., Groenen P.J.F. (1997). Inverse multidimensional scaling. *Journal of Classification*, 12, 3-21.

De Leeuw, J. (2012). On inverse multidimensional scaling. Statistics preprint series, UCLA. [gifi.stat.ucla.edu/janspubs/2012/notes/deleeuw\\_U\\_12b.pdf](http://gifi.stat.ucla.edu/janspubs/2012/notes/deleeuw_U_12b.pdf).

## See Also

[mds](#)

**Examples**

```
## MDS and inverse MDS on kinship data subset
D <- as.matrix(kinshipdelta)[1:6, 1:6]
fit <- mds(D) ## MDS D --> conf
ifit <- inverseMDS(fit$conf) ## inverse MDS conf --> D

## MDS fit on dissimilarity matrix obtained by inverse MDS
op <- par(mfrow = c(3,3))
plot(fit, main = "Original MDS")
for (i in 1:length(ifit)) {
  fit.i <- mds(ifit[[i]])
  plot(fit.i, main = paste0("Inverse MDS (",i, ")"))
}
par(op)
```

---

jackknife

*SMACOF Jackknife*


---

**Description**

These methods perform a SMACOF Jackknife and plot the corresponding solution.

**Usage**

```
## S3 method for class 'smacofB'
jackknife(object, eps = 1e-6, itmax = 100, verbose = FALSE)

## S3 method for class 'smacofJK'
plot(x, plot.dim = c(1,2), hclpar = list(start = 30, end = 200),
plot.lines = TRUE, main, xlab, ylab, xlim, ylim, ...)
```

**Arguments**

object	Object of class "smacofB", i.e., an MDS solution from <code>smacofSym()</code>
itmax	Maximum number of iterations
eps	Convergence criterion
verbose	If TRUE, intermediate stress is printed out
x	Object of class "smacofJK"
plot.dim	Vector with dimensions to be plotted.
hclpar	Parameters to be used for HCL colors (further details see <a href="#">rainbow_hcl</a> )
plot.lines	If TRUE, the Jackknife configurations are plotted are connected with their centroid
main	Plot title.

xlab	Label of x-axis.
ylab	Label of y-axis.
xlim	Scale x-axis.
ylim	Scale y-axis.
...	Further plot arguments passed: see <a href="#">plot</a> for detailed information.

### Details

In order to examine the stability solution of an MDS, a Jackknife on the configurations can be performed (see de Leeuw & Meulman, 1986) and plotted. The plot shows the jackknife configurations which are connected to their centroid. In addition, the original smacof configuration (transformed through Procrustes) is plotted. The Jackknife function itself returns also a stability measure (as ratio of between and total variance), a measure for cross validity, and the dispersion around the original smacof solution.

### Value

smacof.conf	SMACOF configurations
jackknife.conf	An array of n-1 configuration matrices for each Jackknife MDS solution
comparison.conf	Centroid Jackknife configurations (comparison matrix)
stab	Stability measure
cross	Cross validity
disp	Dispersion
loss	Value of the loss function
ndim	Number of dimensions
call	Model call
niter	Number of iterations
nobj	Number of objects

### Author(s)

Jan de Leeuw and Patrick Mair

### References

De Leeuw, J., & Meulman, J. (1986). A special jackknife for multidimensional scaling. *Journal of Classification*, 3, 97-112.

### See Also

[smacofSym](#), [smacofConstraint](#), [plot.smacof](#)

## Examples

```
## symmetric smacof
data(kinshipdelta)
res.smacof <- smacofSym(kinshipdelta)
res.jk <- jackknife(res.smacof)
res.jk

plot(res.jk)
plot(res.jk, hclpar = list(10, 500))
plot(res.jk, hclpar = list(10, 300), plot.lines = FALSE)
```

---

kinshipdelta

*Kinship Terms*

---

## Description

Percentages of how often 15 kinship terms were not grouped together by college students including three external scales.

## Usage

```
data(kinshipdelta)

data(kinshipscales)
```

## Format

Dissimilarity matrix of 15 kinship terms and data frame with the following external scales:  
Gender (1 = male, 2 = female, 9 = missing)  
Generation (-2 = two back, -1 = one back, 0 = same generation, 1 = one ahead, 2 = two ahead)  
Degree (1 = first, 2 = second, 3 = third, 4 = fourth)

## References

Rosenberg, S. & Kim, M. P. (1975). The method of sorting as a data gathering procedure in multivariate research. *Multivariate Behavioral Research*, 10, 489-502.

## Examples

```
data(kinshipdelta)
data(kinshipscales)
```

---

LawLer

*Management Performance Data*

---

### Description

Performance of managers: 3 criteria ("traits") and 3 methods. Traits: T1 = Quality of output, T2 = Ability to generate output, T3 = Demonstrated effort to perform. Methods: M1 = Rating by superior, M2 = Peer rating, M3 = Self-rating.

### Usage

```
data(Lawler)
```

### Format

Symmetric matrix (trait-method combinations) with inter-correlations.

### References

Lawler, E. E. (1967). Management performance as seen from above, below, and within. In Evaluation of executive performance. Princeton, New Jersey. Educational Testing Service.

### Examples

```
data(Lawler)
```

---

morse

*Morse Code Confusion Data*

---

### Description

Confusion percentages between Morse code signals. The scores are derived from confusion rates on 36 Morse code signals (26 for the alphabet; 10 for the numbers 0,...,9). Each Morse code signal is a sequence of up to five 'beeps'. The beeps can be short (0.05 sec) or long (0.15 sec), and, when there are two or more beeps in a signal, they are separated by periods of silence (0.05 sec).

Rothkopf asked 598 subjects to judge whether two signals, presented acoustically one after another, were the same or not. The values are the average percentages with which the answer 'Same!' was given in each combination of row stimulus  $i$  and column stimulus  $j$ , where either  $i$  or  $j$  was the first signal presented. The values are 1 minus the symmetrized confusion rates and are thus dissimilarities.

### Usage

```
data(morse)
```

**Format**

Similarity matrix of 36 morse codes

**References**

Rothkopf, E. Z. (1957). A measure of stimulus similarity and errors in some paired-associate learning. *Journal of Experimental Psychology*, 53, 94-101.

**Examples**

```
data(morse)
```

---

morsescscales

*Morse Code Confusion Scales*

---

**Description**

Two properties of Morse code signals. Each Morse code signal is a sequence of up to five 'beeps'. The beeps can be short (0.05 sec) or long (0.15 sec), and, when there are two or more beeps in a signal, they are separated by periods of silence (0.05 sec). The two external variables are:

- Signal type:
  - 1: All short beeps
  - 2: More short than long beeps
  - 3: Same short and long beeps
  - 4: More long than short beeps
  - 5: All long beeps
- Signal length (in seconds): 1 = .05, 2 = .15, 3 = .25, 4 = .35, 5 = .45, 6 = .55, 7 = .65, 8 = .85, 9 = .95

**Usage**

```
data(morsescscales)
```

**Format**

Matrix of 36 morse codes by 2 properties. The first column contains the morse code letters.

**References**

Rothkopf, E. Z. (1957). A measure of stimulus similarity and errors in some paired-associate learning. *Journal of Experimental Psychology*, 53, 94-101.

**Examples**

```
data(morsescscales)
```



---

OCP

*Organizational Culture Profile*

---

### Description

Contains similarities (correlations) of 54 OCP (see O'Reilly, Chatman, and Caldwell, 1991) items. The last column contains the facet assigned by Bilsky and Jehn (2002).

### Usage

```
data(OCP)
```

### Format

Data frame with OCP item correlations and facet:

i1-i54: OCP item correlations

facet: factor with facets

### References

Bilsky, W. & Jehn, K. (2002). Organizational Culture and Individual Values: Evidence for a Common Structure. In M. Myrtek (Ed.), *The Person in Biological and Social Context*, pp. 211-228. Goettingen, Germany: Hogrefe Press.

### Examples

```
ocpD <- sim2diss(OCP[,-55])
fit <- mds(ocpD, type = "ordinal")
plot(fit)
```

---

partypref

*Party preferences*

---

### Description

Artificial dataset containing the judges in the rows and the parties in the columns.

### Usage

```
data(partypref)
```

### Format

Matrix of party preferences.

**References**

Borg, I., Groenen, P. J. F., & Mair, P. (2010). *Multidimensionale Skalierung*. Muenchen: Hampp Verlag.

**Examples**

```
data(partypref)
```

---

perception

*Rectangle Perception Data*

---

**Description**

42 subjects are assigned to two groups of 21 persons. 120 stimulus pairs of rectangles are presented. One group judges on a scale from 0 to 9 the perception in terms of width-height (WH), the other group in terms of size-shape (SS).

**Usage**

```
data(perception)
```

**Format**

List of subject dissimilarities for WH and SS group.

**References**

Borg, I. & Leutner, D. (1983). Dimensional models for the perception of rectangles. *Perception and Psychophysics*, 34, 257-269.

**Examples**

```
data(perception)
```

---

permtest                      *SMACOF Permutation*

---

### Description

These methods perform a permutation test for a symmetric or an unfolding SMACOF model.

### Usage

```
## S3 method for class 'smacof'
permtest(object, nrep = 100, verbose = TRUE)
## S3 method for class 'smacofR'
permtest(object, nrep = 100, verbose = TRUE)
## S3 method for class 'smacofPerm'
plot(x, alpha = 0.05, main, xlab, ylab, ...)
```

### Arguments

object	Object of class "smacofB", i.e., an MDS solution from smacofSym()
nrep	Number of permutations
verbose	If TRUE, permutation is printed out
x	Object of class "smacofPerm"
alpha	Alpha level
main	Plot title.
xlab	Label of x-axis.
ylab	Label of y-axis.
...	additional plot arguments

### Details

This routine permutes  $m$  dissimilarity values, where  $m$  is the number of lower diagonal elements in the corresponding dissimilarity matrix. For each sample a symmetric, nonmetric SMACOF of dimension  $ndim$  is computed and the stress values are stored in `stressvec`. Using the fitted stress value, the p-value is computed. Subsequently, the empirical cumulative distribution function can be plotted using the plot method.

### Value

stressvec	Vector containing the stress values of the permutation samples
stress.obs	Stress (observed sample)
pval	Resulting p-value
call	Model call
nrep	Number of permutations
nobj	Number of objects

**Author(s)**

Jan de Leeuw and Patrick Mair

**References**

De Leeuw, J., & Stoop, I. (1984). Upper bounds for Kruskal's stress. *Psychometrika*, 49, 301-402.

**See Also**

[jackknife.smacofB](#), [smacofSym](#), [plot.smacof](#)

**Examples**

```
## symmetric smacof
data(kinshipdelta)
res.smacof <- smacofSym(kinshipdelta, ndim = 1, type = "interval")
res.perm <- permtest(res.smacof)
res.perm
plot(res.perm)

## unfolding
data(breakfast)
res.unfolding <- smacofRect(breakfast, itmax = 5000)
res.perm <- permtest(res.unfolding, nrep = 10)
res.perm
plot(res.perm)
```

---

Plato7

*Plato's Seven Works*

---

**Description**

This dataset contains statistical information about Plato's seven works. The underlying problem to this dataset is the fact that the chronological order of Plato's works is unknown. Scholars only know that Republic was his first work, and Laws his last work. For each work, Cox and Brandwood (1959) extracted the last five syllables of each sentence. Each syllable is classified as long or short which gives 32 types. Consequently, we obtain a percentage distribution across the 32 scenarios for each of the seven works.

**Usage**

```
data(Plato7)
```

**Format**

Data frame containing syllable percentages of Plato's 7 works.

## References

Cox, D. R. & Brandwood, L. (1959). On a discriminatory problem connected with the work of Plato. *Journal of the Royal Statistical Society (Series B)*, 21, 195-200.

## Examples

```
data(Plato7)
```

---

plot.smacof	<i>2D SMACOF plots</i>
-------------	------------------------

---

## Description

These methods provide various 2D plots for SMACOF models.

## Usage

```
## S3 method for class 'smacof'
plot(x, plot.type = "confplot", plot.dim = c(1,2), sphere = TRUE,
      subscale = 1, col = 1, label.conf = list(label = TRUE, pos = 3,
        col = 1, cex = 0.8), shepard.x = NULL, identify = FALSE,
      type = "p", pch = 20, asp = 1, main, xlab, ylab,
      xlim, ylim, ...)

## S3 method for class 'smacofR'
plot(x, plot.type = "confplot", joint = TRUE, plot.dim = c(1,2),
      col.rows = hcl(0), col.columns = hcl(240),
      label.conf.rows = list(label = TRUE, pos = 3,
        col = hcl(0, l = 50), cex = 0.8),
      label.conf.columns = list(label = TRUE, pos = 3,
        col = hcl(240, l = 50), cex = 0.8), type = "p", pch = 20, main,
      xlab, ylab, xlim, ylim, ...)

## S3 method for class 'smacofID'
plot(x, plot.type = "confplot", plot.dim = c(1,2), subscale = 5,
      label.conf = list(label = TRUE, pos = 1, col = 1),
      identify = FALSE, type, main, xlab, ylab, xlim, ylim, ...)
```

## Arguments

x	Object of class "smacof", "smacofR", and "smacofID" (see details)
plot.type	String indicating which type of plot to be produced: "confplot", "resplot", "Shepard", "stressplot", "bubbleplot" (see details)
plot.dim	Vector with dimensions to be plotted.
main	Plot title.
xlab	Label of x-axis.

ylab	Label of y-axis.
xlim	Scale x-axis.
ylim	Scale y-axis.
type	What type of plot should be drawn (see also <a href="#">plot</a> ).
pch	Plot symbol.
asp	Aspect ratio.
col	Point color.
sphere	In case of spherical smacof, whether sphere should be plotted or not.
bubscale	Scaling factor (size) for the bubble plot.
label.conf	List with arguments for plotting the labels of the configurations in a configuration plot (logical value whether to plot labels or not, label position, label color).
shepard.x	Shepard plot only: original data (e.g. correlation matrix) can be provided for plotting on x-axis.
identify	If TRUE, the <code>identify()</code> function is called internally that allows to add configuration labels by mouse click.
joint	If TRUE, the configurations are plotted jointly in rectangular smacof.
col.rows	Row colors in rectangular configuration plot.
col.columns	Column colors in rectangular configuration plot.
label.conf.rows	List with arguments for plotting the labels of the row configurations in a rectangular configuration plot (logical value whether to plot labels or not, label position, label color).
label.conf.columns	List with arguments for plotting the labels of the columns configurations in a rectangular configuration plot (logical value whether to plot labels or not, label position, label color).
...	Further plot arguments passed: see <a href="#">plot</a> for detailed information.

## Details

`smacofSym()` creates object of class "smacof", whereas `smacofRect()` produces "smacofR" and `smacofIndDiff()` generates "smacofID".

Plot description:

- Configuration plot (`plot.type = "confplot"`): Plots the MDS configurations.
- Residual plot (`plot.type = "resplot"`): Plots the normalized dissimilarities (d-hats) distances against the fitted distances.
- Shepard diagram (`plot.type = "Shepard"`): Diagram with the observed dissimilarities against the fitted distances including (isotonic) regression line.
- Stress decomposition plot (`plot.type = "stressplot"`): Plots the stress contribution in of each observation. Note that it rescales the stress-per-point (SPP) from the corresponding smacof function to percentages (sum is 100). The higher the contribution, the worse the fit.

- Bubble plot (`plot.type = "bubbleplot"`, not available for rectangular SMACOF): Combines the configuration plot with the point stress contribution. The larger the bubbles, the better the fit.

For `smacofIndDiff()` the residual plot, Shepard diagram, and stress plot are based on the sum of the residuals across individuals/ways. The configuration plot represents the group stimulus space (i.e., joint configurations).

### See Also

[plot.procr](#)

### Examples

```
## 2D plots for basic SMACOF
data(trading)
res <- smacofSym(trading)
plot(res, plot.type = "confplot")
plot(res, plot.type = "Shepard")
plot(res, plot.type = "stressplot")
plot(res, plot.type = "resplot")
plot(res, plot.type = "bubbleplot")

data(wish)
wish.dist <- sim2diss(wish, method = 7)
res <- smacofSym(wish.dist, type = "ordinal")
res
plot(res, ylim = c(-0.6, 1)) ## aspect ratio = 1 (default)
plot(res, label.conf = list(FALSE), asp = NA) ## asp not 1, no labels
plot(res, type = "n") ## labels only (at configuration coordinates)

## Shepard plots
ekmanD <- sim2diss(ekman)
fit1 <- mds(ekmanD, type = "ordinal")
plot(fit1, plot.type = "Shepard")
plot(fit1, plot.type = "Shepard", shepard.x = ekman) ## original data

## Joint configuration plot and row/column stressplots for rectangular SMACOF
data(breakfast)
res <- smacofRect(breakfast)
plot(res, plot.type = "confplot")

plot(res, plot.type = "stressplot")

plot(res, type = "p", pch = 25)

plot(res, type = "p", pch = 25, col.columns = 3,
label.conf.columns = list(label = TRUE, pos = 3, col = 3), col.rows = 8,
label.conf.rows = list(label = TRUE, pos = 3, col = 8))

plot(res, joint = TRUE, type = "p", pch = 25, col.columns = 4,
label.conf.columns = list(label = TRUE, pos = 3, col = 4),
col.rows = 8, label.conf.rows = list(label = TRUE, pos = 3, col = 8))
```

Procrustes

*Procrustean Similarity Transformations***Description**

Solves the Procrustean problem of fitting one (MDS) configuration (testee) to another (target) MDS configuration.

**Usage**

```
Procrustes(X, Y)
```

```
## S3 method for class 'procr'
plot(x, plot.type = "jointplot", plot.dim = c(1,2), main, xlab, ylab,
      xlim, ylim, asp = 1, pch = 20, col.X = "cadetblue", col.Y = "gray",
      col.Yhat = "coral1",
      label.conf = list(label = TRUE, pos = 3, cex = 0.8),
      arrows = TRUE, length = 0.10, ...)
```

**Arguments**

X	Target configuration
Y	Testee configuration
x	Object of class procr
plot.type	Either "jointplot" or "transplot"
plot.dim	Vector with dimensions to be plotted.
main	Plot title.
xlab	Label of x-axis.
ylab	Label of y-axis.
xlim	Scale x-axis.
ylim	Scale y-axis.
pch	Plot symbol.
asp	Aspect ratio.
col.X	Color target configuration.
col.Y	Color testee configuration.
col.Yhat	Color transformed configuration.
label.conf	List with arguments for plotting the labels of the configurations in a configuration plot (logical value whether to plot labels or not, label position, label color).
length	length of the edges of the arrow head (in inches).
arrows	For "transplot" only, whether arrows should be plotted or not
...	Additional plot arguments.



**Details**

Y is going to be modified by finding an optimal dilation factor, an optimal translation and rotation for Y such that it is as similar as possible to X. X remains untouched.

**Value**

Returns an object of class `procr` with:

X	Input target configuration
Y	Input testee configuration
Yhat	Procrustes transformed (fitted) configuration
translation	Translation vector
dilation	Dilation factor
rotation	Rotation-reflection matrix
confdistX	Configuration distances X
confdistY	Configuration distances Y
confdistYhat	Configuration distances of fitted configuration
congcoef	Congruence coefficient

**References**

Borg, I., & Groenen, P. J. F. (2005). *Modern Multidimensional Scaling* (2nd ed.). Springer.

**Examples**

```
## artificial example:
X <- matrix(c(1, -1, -1, 1, 2, 2, -2, -2), ncol = 2)
Y <- matrix(c(0.07, 0.93, 1.93, 1.07, 2.62, 3.12, 1.38, 0.88), ncol = 2)
op <- par(mfrow = c(1,2))
plot(X[,1], X[,2], xlim = c(-3, 3), ylim = c(-2, 3.5), asp = 1, xlab = "", ylab = "")
rect(-1, -2, 1, 2)
points(Y[,1], Y[,2], xlim = c(-3, 3), col = "gray")
polygon(Y[,1], Y[,2], border = "gray")

fitp <- Procrustes(X, Y)
plot(fitp$Yhat[,1], fitp$Yhat[,2], col = "red", xlim = c(-3, 3), ylim = c(-2, 3.5),
     asp = 1, xlab = "", ylab = "")
polygon(fitp$Yhat[,1], fitp$Yhat[,2], border = "red")
par(op)

## MDS example:
eastD <- sim2diss(EW_eng$east)
attr(eastD, "Labels") <- abbreviate(attr(eastD, "Labels"))
fit.east <- mds(eastD, type = "ordinal")
westD <- sim2diss(EW_eng$west)
attr(westD, "Labels") <- abbreviate(attr(westD, "Labels"))
fit.west <- mds(westD, type = "ordinal", init = torgerson(eastD))
```

```

fit.proc <- Procrustes(fit.east$conf, fit.west$conf)
fit.proc

## Configuration plots; Procrustes plots.
op <- par(mfrow = c(2,2))
plot(fit.east, main = "MDS East Germany")
plot(fit.west, main = "MDS West Germany")
plot(fit.proc)      ## X (blue) and Yhat (red)
plot(fit.proc, plot.type = "transplot", length = 0.05) ## Y (gray) and Yhat (red)
par(op)

```

---

PVQ40

*Portrait Value Questionnaire*


---

### Description

The PVQ40 (Schwartz et al., 1999) consists of 40 items, each a short portrait of one person. For example, to measure power, the PVQ includes two portraits (male/female versions): It is important to him to be rich. He wants to have a lot of money and expensive things. It is important to him to get respect from others. He wants people to do what he says. Respondent indicates on 6-point bipolar rating scale (1 ... not at all like me, 6 ... very much like me) the degree to which the description also fits him-/herself:

### Usage

```
data(PVQ40)
```

### Format

PVQ40 data of 151 adults from various states in the USA:

sd1-sd4: self-direction

po1-po3: power

un1-un6: universalism

ac1-ac4: achievement

se1-se5: security

st1-st3: stimulation

co1-co4: conformity

tr1-tr4: tradition

he1-he3: hedonism

be1-be4: benevolence

### References

Borg, I., & Groenen, P. J. F. (2005). *Modern Multidimensional Scaling* (2nd ed.). Springer.

**Examples**

```
pvqdiss <- sim2diss(cor(PVQ40, use = "complete.obs"))
fit <- mds(pvqdiss, type = "ordinal")
plot(fit)
plot(fit, plot.type = "Shepard")
```

---

randomstress

*Stress Calculation for Random Dissimilarities*

---

**Description**

Creates random dissimilarity matrices (n objects), fits an MDS, and returns the stress values of each MDS fit.

**Usage**

```
randomstress(n, ndim, nrep, type = c("ratio", "interval", "ordinal", "mspline"))
```

**Arguments**

n	Number of objects
ndim	Number of dimensions for MDS
nrep	Number of random samples
type	MDS type

**Details**

The random dissimilarities are drawn from a U(0,1) distribution.

**Value**

Returns a vector with stress values.

**References**

Spence I., Ogilvie, J.C. (1973). A table of expected stress values for random rankings in nonmetric multidimensional scaling. *Multivariate Behavioral Research*, 8, 511-517.

**Examples**

```
## 9 objects, 2 dimensions, interval MDS (100 replications)
stressvec <- randomstress(n = 8, ndim = 2, nrep = 100, type = "interval")
mean(stressvec)
```

---

rectangles	<i>Rectangles</i>
------------	-------------------

---

**Description**

These data are based on an experiment by Borg and Leutner (1983). They constructed rectangles based on a specific grid structure. The size of the rectangles is given in the constraints matrix. In total, we have 16 rectangles. 21 subjects had to rate twice the similarity of each pair of rectangles on a scale from 0 (identical) to 9 (very different). The dist object contains the average ratings which are dissimilarities.

**Usage**

```
data(rectangles)
data(rect_constr)
```

**Format**

The rectangles are object of class `dist`, the constraints are given as matrix

**References**

Borg, I., & Leutner, D. (1983). Dimensional models for the perception of rectangles. *Perception and Psychophysics*, 34, 257-269.

Borg, I., Groenen, P. J. F., & Mair, P. (2010). *Multidimensionale Skalierung*. Muenchen: Hampp Verlag.

**Examples**

```
data(rectangles)
data(rect_constr)
```

---

residuals.smacof	<i>Residuals</i>
------------------	------------------

---

**Description**

Computes the residuals by subtracting the configuration dissimilarities from the observed dissimilarities.

**Usage**

```
## S3 method for class 'smacof'
residuals(object, ...)
## S3 method for class 'smacofR'
residuals(object, ...)
## S3 method for class 'smacofID'
residuals(object, ...)
```

**Arguments**

object	Object of class smacof, smacofR (rectangular), or smacofID (individual differences)
...	Ignored

**Examples**

```
data(kinshipdelta)
res <- smacofSym(kinshipdelta)
residuals(res)
```

---

RockHard

*RockHard Ratings*

---

**Description**

Data from RockHard Magazine: In this German Heavy Metal Magazine around 50 records are rated by the writers on a scale from (0 ... worst to 10 ... best) each month. The dataset contains all ratings from 2013.

**Usage**

```
data(RockHard)
```

**Format**

Data frame with raters in the columns, bands/albums in the rows.

**References**

Mair, P., de Leeuw, J., & Wurzer, M. (2015). Multidimensional Unfolding. Wiley StatsRef: Statistics Reference Online. New York: Wiley.

**Examples**

```
head(RockHard)
```

---

sim2diss	<i>Converts similarites to dissimilarities</i>
----------	--

---

### Description

Utility function for converting similarities into dissimilarities. Different methods are provided.

### Usage

```
sim2diss(similmat, method = "corr", to.dist = TRUE)
```

### Arguments

similmat	Similarity matrix (not necessarily symmetric, nor quadratic)
method	Various methods for converting similarities into dissimilarities: "corr", "neglog", "counts", or an integer value (see details)
to.dist	If TRUE, object of class dist is produced

### Details

We provide the following methods for converting similarities  $S$  into dissimilarities  $D$ : "corr" is suited for correlation matrices and takes  $D = \sqrt{1 - S}$ . "neglog" takes the negative logarithm in terms of  $-\log(S)$ . Having frequencies, "counts" is appropriate which does  $-\log((S[i, j] * S[j, i]) / (S[i, i] * S[j, j]))$ . The user can specify also an integer value  $v$ . In this case `sim2diss()` computes  $v - S$ .

### Value

Returns dissimilarities either as matrix or as dist object.

### Examples

```
## Converting Ekman data (similarities) into dissimilarities by subtraction from 1
data(ekman)
ekman.diss <- sim2diss(ekman, method = 1)
res <- smacofSym(ekman.diss)
```

---

smacofConstraint	<i>SMACOF Constraint</i>
------------------	--------------------------

---

## Description

SMACOF with internal constraints on the configurations.

## Usage

```
smacofConstraint(delta, constraint = "linear", external, ndim = 2,
  type = c("ratio", "interval", "ordinal", "mspline"), weightmat = NULL,
  init = NULL, ties = "primary", verbose = FALSE, modulus = 1,
  itmax = 1000, eps = 1e-6, spline.intKnots = 4, spline.degree = 2,
  constraint.type = c("ratio", "interval", "ordinal", "spline",
    "mspline"), constraint.ties = "primary",
  constraint.spline.intKnots = 2, constraint.spline.degree = 2)
```

## Arguments

delta	Either a symmetric dissimilarity matrix or an object of class "dist"
constraint	Type of constraint: "linear", "unique", "diagonal", or a user-specified function (see details)
external	Data frame or matrix with external covariates, or list for simplex and circumplex (see details)
ndim	Number of dimensions
type	MDS type: "interval", "ratio", "ordinal" (nonmetric MDS), or "mspline"
weightmat	Optional matrix with dissimilarity weights
init	Optional matrix with starting values for configurations
ties	Tie specification for non-metric MDS only: "primary", "secondary", or "tertiary"
verbose	If TRUE, intermediate stress is printed out
modulus	Number of smacof iterations per monotone regression call
itmax	Maximum number of iterations
eps	Convergence criterion
spline.degree	Degree of the spline for "mspline" MDS type
spline.intKnots	Number of interior knots of the spline for "mspline" MDS type
constraint.type	Transformation for external covariates: "ratio", "interval", "ordinal", "spline", or "mspline")
constraint.ties	Tie specification for external covariates with constraint.type = "ordinal": "primary", "secondary", or "tertiary"

```

constraint.spline.intKnots
    Number of interior knots for external covariates with constraint.type = "spline"
    or "mspline"
constraint.spline.degree
    Degree of the spline for external covariates with constraint.type = "spline"
    or "mspline"

```

## Details

The argument `external` is mandatory and allows for the specification of a covariate data frame (or matrix) of dimension  $(n \times q)$ . Alternatively, for simplex fitting the user can specify a list of the following structure: `external = list("simplex", dim2)` with `dim2` denoting the dimension of the simplex with  $\text{dim2} < n$ . For a circumplex fitting, the list has to be of the following form: `external = list("circumplex", dim2, k1, k2)` with  $1 \leq k1 \leq k2 \leq n$  (see also examples section). `k1` and `k2` denote the circumplex width.

In constraint `smacof`, the configuration matrix is subject of a constraint based on the external scales (predictors). This constraint can be specified using the `constraint` argument. We provide the following standard setting:

For `constraint = "linear"` the configurations  $X$  are decomposed linearly, i.e.  $X = ZC$  where  $Z$  is the known predictor matrix specified using `external`.

The same for `constraint = "diagonal"` where  $X$  needs to be of dimension  $(n \times q)$  where  $q$  is the number of columns of the external scale matrix (and thus number of dimensions). Here,  $C$  is restricted to be diagonal.

For `constraint = "linear"` or `"diagonal"`, the external covariates  $Z$  can be optimally transformed as specified by `constraint.type`. Choosing the number of covariates equal to the number of dimensions together with `constraint.type = "ordinal"`, `constraint.ties = "primary"` will effectively restrict the configuration to parallel regions defined by the categories of the covariates. Note that missing values of the covariates are estimated by the model.

For `constraint = "unique"` we get the Bentler-Weeks uniqueness model. Hence  $X$  is of dimension  $(n \times (n + p))$ . This implies that we fit a certain number of dimensions  $p$  and, in addition we extract  $n$  additional dimensions where each object is scored on a separate dimension. More technical details can be found in the corresponding JSS article (reference see below).

In addition, the user can specify his own constraint function with the following arguments: configuration matrix with starting values (`init`) (mandatory in this case), matrix  $V$  (`weightmat`; based on the weight matrix, see package vignette), external scale matrix (`external`). The function must return a matrix of resulting configurations.

## Value

<code>delta</code>	Observed dissimilarities
<code>obsdiss</code>	Observed dissimilarities, normalized
<code>confdiss</code>	Configuration dissimilarities
<code>conf</code>	Matrix of final configurations
<code>C</code>	Matrix with restrictions
<code>stress</code>	Stress-1 value



spp	Stress per point
ndim	Number of dimensions
extvars	List for each external covariate with a list of class "optscal"
model	Type of smacof model
niter	Number of iterations
nobj	Number of objects

**Author(s)**

Jan de Leeuw and Patrick Mair

**References**

- de Leeuw, J. & Mair, P. (2009). Multidimensional scaling using majorization: The R package smacof. *Journal of Statistical Software*, 31(3), 1-30, <http://www.jstatsoft.org/v31/i03/>
- de Leeuw, J., & Heiser, W. (1980). Multidimensional scaling with restrictions on the configurations. In P. R. Krishnaiah (eds.), *Multivariate Analysis V*, pp. 501-522. North-Holland.

**See Also**

[smacofSym](#), [smacofRect](#), [smacofIndDiff](#), [smacofSphere](#)

**Examples**

```
## SMACOF with linear configuration constraints
data(kinshipdelta)
data(kinshipscales)
res.lin1 <- smacofConstraint(kinshipdelta, constraint = "linear", external = kinshipscales)

## X = ZC decomposition
X <- res.lin1$conf          ## resulting configurations X
Z <- as.matrix(kinshipscales)
C <- res.lin1$C
C
Z%*%C                      ## check: should be equal to X

## SMACOF with diagonal constraints
res.diag <- smacofConstraint(kinshipdelta, constraint = "diagonal",
external = kinshipscales, ndim = 3)

## linear constraints, no optimal scaling
res.lin <- smacofConstraint(morse, type = "ordinal", ties = "primary",
constraint = "linear",
external = morsescscales[,2:3])

## linear constraints, optimal scaling (regional MDS)
res.parreg <- smacofConstraint(morse, type = "ordinal", ties = "primary",
constraint = "linear",
external = morsescscales[,2:3],
```

```

constraint.type = "ordinal")
plot(res.parreg)

## SMACOF with unique constraints (Bentler-Weeks model)
res.unique <- smacofConstraint(kinshipdelta, constraint = "unique",
                             external = kinshipscales)
plot(res.unique)

## Fitting a simplex with q = 4 (i.e., n-1), diagonal constraints
res.simp <- smacofConstraint(kinshipdelta, constraint = "diagonal",
                            external = list("simplex", 4), ndim = 3)
plot(res.simp)

## Fitting a circumplex with q = 3, k1 = 1, k2 = 2, diagonal constraints
res.circ <- smacofConstraint(kinshipdelta, constraint = "diagonal",
                            external = list("circumplex", 3, 1, 2), ndim = 3)
plot(res.circ)

```

---

smacofIndDiff

*SMACOF for Individual Differences*


---

## Description

Performs smacof for individual differences also known as Three-Way smacof on a list of dissimilarity matrices. Various restrictions decompositions and restrictions on the weight matrix are provided. The most prominent models are INDSCAL and IDIOSCAL.

## Usage

```

smacofIndDiff(delta, ndim = 2, type = c("ratio", "interval", "ordinal"),
              constraint = c("indscal", "idioscal", "identity"),
              weightmat = NULL, init = NULL, ties = "primary",
              verbose = FALSE, modulus = 1, itmax = 1000, eps = 1e-6)

indscal(delta, ndim = 2, type = c("ratio", "interval", "ordinal"),
        weightmat = NULL, init = NULL, ties = "primary",
        verbose = FALSE, modulus = 1, itmax = 1000, eps = 1e-6)

idioscal(delta, ndim = 2, type = c("ratio", "interval", "ordinal"),
         weightmat = NULL, init = NULL, ties = "primary",
         verbose = FALSE, modulus = 1, itmax = 1000, eps = 1e-6)

```

## Arguments

delta	A list of dissimilarity matrices or a list objects of class dist
ndim	Number of dimensions

type	MDS type
weightmat	Optional matrix with dissimilarity weights
init	Matrix with starting values for configurations (optional)
ties	Tie specification for non-metric MDS
constraint	Either "indscal", "idioscal", or "identity" (see details)
verbose	If TRUE, intermediate stress is printed out
modulus	Number of smacof iterations per monotone regression call
itmax	Maximum number of iterations
eps	Convergence criterion

### Details

If the constraint is "indscal", INDSCAL is performed with configuration weight matrices restricted to be diagonal. `indscal()` is a corresponding wrapper function that can be used instead of `smacofIndDiff()` with "indscal" constraints.

IDIOSCAL can be computed using the "idioscal" argument. The weight matrices are then unconstrained. `idioscal()` is a corresponding wrapper function that can be used instead of `smacofIndDiff()` with "idioscal" constraints.

Additional weight restrictions can be imposed with "identity" which restricts the configurations across individuals/replications/ways to be equal.

### Value

delta	Observed dissimilarities
obsdiss	List of observed dissimilarities, normalized
confdiss	List of configuration dissimilarities
conf	List of matrices of final configurations
gspace	Joint configurations aka group stimulus space
cweights	Configuration weights
stress	Stress-1 value
spp	Stress per point
sps	Stress per subject (matrix)
ndim	Number of dimensions
model	Type of smacof model
niter	Number of iterations
nobj	Number of objects

### Author(s)

Jan de Leeuw and Patrick Mair

## References

de Leeuw, J., & Mair, P. (2009). Multidimensional scaling using majorization: The R package smacof. *Journal of Statistical Software*, 31(3), 1-30, <http://www.jstatsoft.org/v31/i03/>

## See Also

[smacofConstraint](#), [smacofSym](#), [smacofRect](#), [smacofSphere](#)

## Examples

```
data(perception)

res.id <- smacofIndDiff(perception, constraint = "identity")  ## identity restricted weights
summary(res.id)
res.id$cweights
plot(res.id)
plot(res.id, type = "p", pch = 25, col = 4, label.conf = list(label = TRUE, pos = 3, col = 4))

res.diag <- smacofIndDiff(perception, constraint = "indscal") ## diagonally restricted weights
res.diag$cweights
plot(res.diag)
plot(res.diag, type = "p", pch = 25, col = 4, label.conf = list(label = TRUE, pos = 3, col = 4))

res.idio <- smacofIndDiff(perception, constraint = "idioscal") ## IDIOSCAL
Wk <- res.idio$cweights
G <- res.idio$gspace
G
G

data(helm)
res.helm <- smacofIndDiff(helm, constraint = "indscal")
plot(res.helm, plot.type = "confplot")
barplot(sort(res.helm$sps, decreasing = TRUE), main = "Stress per Subject", cex.names = 0.8)
plot(res.helm, plot.type = "bubbleplot")
plot(res.helm, plot.type = "stressplot")
plot(res.helm, plot.type = "Shepard")
plot(res.helm, plot.type = "resplot")

## idioscal and indscal with starting configurations:
startconf <- matrix(rnorm(20), 10, 2)
idioscal(helm, init = startconf)
indscal(helm, init = startconf)
```

**Description**

Variant of smacof for rectangular matrices (typically ratings, preferences) which is also known as metric unfolding.

**Usage**

```
smacofRect(delta, ndim = 2, circle = c("none", "row", "column"), weightmat = NULL,
           init = NULL, verbose = FALSE, itmax = 1000, reg = 1e-6, eps = 1e-6)
```

```
unfolding(delta, ndim = 2, circle = c("none", "row", "column"), weightmat = NULL,
          init = NULL, verbose = FALSE, itmax = 1000, reg = 1e-6, eps = 1e-6)
```

**Arguments**

delta	Data frame or matrix of preferences, ratings, dissimilarities.
ndim	Number of dimensions
circle	If "column", the column configurations are restricted to be on a circle, if "row", row configurations are on a circle, if "none", there are no restrictions on row and column configurations
weightmat	Optional matrix with dissimilarity weights
init	Matrix with starting values for configurations (optional)
verbose	If TRUE, intermediate stress is printed out
itmax	Maximum number of iterations
reg	Regularization factor, prevents distances from being 0
eps	Convergence criterion

**Details**

Creates an object of class smacofR. The unfolding() function is a wrapper function and can be used instead of smacofRect().

**Value**

obsdiss	Observed dissimilarities, corresponds to delta
confdiss	Configuration dissimilarities
conf.row	Matrix of final row configurations
conf.col	Matrix of final column configurations
stress	Final, normalized stress value
spp.row	Stress per point, rows
spp.col	Stress per point, columns
congvec	Vector of congruency coefficients
ndim	Number of dimensions
model	Type of smacof model

niter	Number of iterations
nind	Number of individuals (rows)
nobj	Number of objects (columns)

**Author(s)**

Jan de Leeuw and Patrick Mair

**References**

de Leeuw, J. & Mair, P. (2009). Multidimensional scaling using majorization: The R package smacof. *Journal of Statistical Software*, 31(3), 1-30, <http://www.jstatsoft.org/v31/i03/>

**See Also**

[smacofConstraint](#), [smacofSym](#), [smacofIndDiff](#), [smacofSphere](#)

**Examples**

```
res <- unfolding(breakfast)
res
summary(res)

## various configuration plots
plot(res)
plot(res, type = "p", pch = 25)
plot(res, type = "p", pch = 25, col.columns = 3,
label.conf.columns = list(label = TRUE, pos = 3, col = 3),
col.rows = 8, label.conf.rows = list(label = TRUE, pos = 3, col = 8))
```

---

smacofSphere

*Spherical SMACOF*

---

**Description**

Dual and primal approach for spherical SMACOF.

**Usage**

```
smacofSphere(delta, algorithm = c("dual", "primal"), ndim = 2,
type = c("ratio", "interval", "ordinal"), weightmat = NULL, init = NULL,
ties = "primary", verbose = TRUE, penalty = 100, relax = FALSE,
modulus = 1, itmax = 1000, eps = 1e-6)
```

**Arguments**

delta	Either a symmetric dissimilarity matrix or an object of class <code>dist</code>
algorithm	Algorithm type
penalty	Penalty parameter for dual algorithm (larger 0)
ndim	Number of dimensions
weightmat	Optional matrix with dissimilarity weights
init	Matrix with starting values for configurations (optional)
type	MDS type: "interval", "ratio", or "ordinal" (nonmetric MDS)
ties	Tie specification for non-metric MDS only
verbose	If TRUE, intermediate stress is printed out
relax	If TRUE, block relaxation is used for majorization (dual algorithm)
modulus	Number of smacof iterations per monotone regression call
itmax	Maximum number of iterations
eps	Convergence criterion

**Value**

delta	Observed dissimilarities
obsdiss	Observed dissimilarities, normalized
obsdiss1	Dual SMACOF: Observed dissimilarities
obsdiss2	Dual SMACOF: Restriction matrix
confdiss	Configuration dissimilarities
conf	Matrix with fitted configurations
spp	Stress per point
stress	Stress-1 value
ndim	Number of dimensions
dummyvec	Dummy vector of restriction matrix
model	Type of smacof model
niter	Number of iterations
nobj	Number of objects

**Author(s)**

Jan de Leeuw and Patrick Mair

**References**

De Leeuw, J. & Mair, P. (2009). Multidimensional scaling using majorization: The R package smacof. *Journal of Statistical Software*, 31(3), 1-30, <http://www.jstatsoft.org/v31/i03/>

**See Also**

[smacofRect](#), [smacofIndDiff](#), [smacofSym](#), [smacofConstraint](#)

**Examples**

```
## spherical SMACOF solution for trading data
data(trading)
res <- smacofSphere(trading)
res
summary(res)
```

---

smacofSym

*Symmetric smacof*

---

**Description**

Multidimensional scaling on a symmetric dissimilarity matrix using SMACOF.

**Usage**

```
smacofSym(delta, ndim = 2, type = c("ratio", "interval", "ordinal", "mspline"),
           weightmat = NULL, init = NULL, ties = "primary", verbose = FALSE,
           relax = FALSE, modulus = 1, itmax = 1000, eps = 1e-06,
           spline.degree = 2, spline.intKnots = 2)
```

```
mds(delta, ndim = 2, type = c("ratio", "interval", "ordinal", "mspline"),
     weightmat = NULL, init = NULL, ties = "primary", verbose = FALSE,
     relax = FALSE, modulus = 1, itmax = 1000, eps = 1e-06,
     spline.degree = 2, spline.intKnots = 2)
```

**Arguments**

delta	Either a symmetric dissimilarity matrix or an object of class "dist"
ndim	Number of dimensions
weightmat	Optional matrix with dissimilarity weights
init	Matrix with starting values for configurations (optional)
type	MDS type: "interval", "ratio", "ordinal" (nonmetric MDS), or "mspline"
ties	Tie specification for ordinal MDS only: "primary", "secondary", or "tertiary"
verbose	If TRUE, intermediate stress is printed out
relax	If TRUE, block relaxation is used for majorization
modulus	Number of smacof iterations per monotone regression call
itmax	Maximum number of iterations



eps	Convergence criterion
spline.degree	Degree of the spline for "mspline" MDS type
spline.intKnots	Number of interior knots of the spline for "mspline" MDS type

### Details

This is the simplest MDS-SMACOF version of the package. It solves the stress target function for symmetric dissimilarities by means of the majorization approach (SMACOF) and reports the Stress-1 value (normalized). The main output are the coordinates in the low-dimensional space (configurations; conf).

The function `mds()` is a wrapper function and can be used instead of `smacofSym()`

This function allows for fitting three basic types of MDS: ratio MDS (default), interval MDS (polynomial transformation), and ordinal MDS (aka nonmetric MDS). It also returns the point stress, i.e. the larger the contribution of a point to the total stress, the worse the fit (see also [plot.smacof](#)).

### Value

delta	Observed dissimilarities, not normalized
dhat	Disparities (transformed proximities, approximated distances, d-hats)
confdiss	Configuration distances
conf	Matrix of fitted configurations
stress	Stress-1 value
spp	Stress per point (stress contribution in percentages)
resmat	Matrix with squared residuals.
weightmat	Weight matrix.
ndim	Number of dimensions
model	Name of smacof model
niter	Number of iterations
nobj	Number of objects
type	Type of MDS model

### Author(s)

Jan de Leeuw and Patrick Mair

### References

- De Leeuw, J. & Mair, P. (2009). Multidimensional scaling using majorization: The R package smacof. *Journal of Statistical Software*, 31(3), 1-30, <http://www.jstatsoft.org/v31/i03/>
- Borg, I., & Groenen, P. J. F. (2005). *Modern Multidimensional Scaling* (2nd ed.). Springer.
- Borg, I., Groenen, P. J. F., & Mair, P. (2013). *Applied Multidimensional Scaling*. Springer.

**See Also**

[smacofConstraint](#), [smacofRect](#), [smacofIndDiff](#), [smacofSphere](#), [plot.smacof](#)

**Examples**

```
## simple SMACOF solution for kinship data
res <- mds(kinshipdelta)
res
summary(res)
plot(res)
plot(res, type = "p", label.conf = list(label = TRUE, col = "darkgray"), pch = 25, col = "red")

## interval MDS
res <- smacofSym(kinshipdelta, type = "interval")
res

## 3D nonmetric SMACOF solution for trading data (secondary approach to ties)
data(trading)
res <- smacofSym(trading, ndim = 3, type = "ordinal", ties = "secondary")
res
```

---

stardist

*Distances among stars in zodiac signs*


---

**Description**

A distance matrix for the 10 brightest stars in each of the 12 zodiac signs was computed. Astronomers measure the projected positions of objects on the celestial sphere in two angles, i.e. right ascension  $\alpha$  and declination  $\delta$ . For every zodiac sign, the projected distances on the sky between individual stars  $S_i$  and  $S_j$  have been calculated in decimal degrees by means of the Pythagorean theorem

$$d_{i,j} = \sqrt{(\alpha_i - \alpha_j)^2 + (\delta_i - \delta_j)^2}$$

assuming planar geometry. Since the zodiac signs are relatively small compared to the whole celestial sphere and the computation is only done for illustrative purposes, such a simplified assumption is appropriate.

**Usage**

```
data(stardist)
```

**Format**

A dist object containing the star distances.

**Note**

Thanks to Paul Eigenthaler, Department of Astronomy, University of Vienna for calculating the distances.

**Examples**

```
data(stardist)
```

---

summary.smacofB	<i>S3 methods for smacof</i>
-----------------	------------------------------

---

**Description**

Print and summary methods for objects of class smacofB, smacofR (rectangular), and smacofID (individual differences).

**Usage**

```
## S3 method for class 'smacofB'  
summary(object, ...)  
## S3 method for class 'smacofB'  
print(x, ...)  
## S3 method for class 'smacofR'  
summary(object, ...)  
## S3 method for class 'smacofR'  
print(x, ...)  
## S3 method for class 'smacofID'  
summary(object, ...)  
## S3 method for class 'smacofID'  
print(x, ...)
```

**Arguments**

object	Object of class smacofB, smacofR, smacofID
x	Object of class smacofB, smacofR, smacofID
...	Ignored

**Examples**

```
data(kinshipdelta)  
res <- smacofSym(kinshipdelta)  
res  
summary(res)
```

---

torgerson

*Torgerson Scaling*

---

**Description**

Classical MDS aka Torgerson Scaling

**Usage**

```
torgerson(delta, p)
```

**Arguments**

delta	Dissimilarity matrix
p	Number of dimensions

**Value**

Returns an  $n \times p$  matrix of configurations

**References**

Borg, I., & Groenen, P. J. F. (2005). *Modern Multidimensional Scaling* (2nd ed.). Springer.

**Examples**

```
fit <- torgerson(Guerry)
```

---

trading

*Trading data*

---

**Description**

Data from the *New Geographical Digest* (1986), analysed in Cox and Cox (2001), on which countries traded with other countries. For 20 countries the main trading partners are dichotomously scored (1 = trade performed, 0 = trade not performed). Based on this dichotomous matrix the dissimilarities are computed using the Jaccard coefficient.

**Usage**

```
data(trading)
```

**Format**

Object of class "dist" with dissimilarities of the following countries:

Arge: Argentina

Aust: Australia

Braz: Brazil

Cana: Canada

Chin: China

Czec: Czechoslovakia

Egyp: Egypt

E.Ge: East Germany

Fran: France

Hung: Hungary

Indi: India

Ital: Italy

Japa: Japan

N.Ze: New Zealand

Pola: Poland

Swed: Sweden

USA

USSR: Soviet Union

U.K.: United Kingdom

W.Ge: West Germany

**References**

Cox, T.F., Cox, M.A.A. (1991). Multidimensional scaling on a sphere. *Communications in Statistics: Theory and Methods*, 20, 2943-2953.

**Examples**

```
data(trading)
```

---

 uniscale *Unidimensional Scaling*


---

**Description**

Simple implementation where all dissimilarity permutations are subject to a 1D MDS fit and the one which leads to a minimal stress values is returned.

**Usage**

```
uniscale(delta, weightmat = NULL)
```

**Arguments**

delta	Either a symmetric dissimilarity matrix or an object of class "dist"
weightmat	Optional matrix with dissimilarity weights

**Value**

delta	Observed dissimilarities, not normalized
confdiss	Configuration distances
conf	Vector with fitted configurations
stress	Stress-1 value
weightmat	Weight matrix
nobj	Number of objects
npermtot	Total number of permutations (factorial)
npermscale	Number of accepted permutations (monotonicity check)

**References**

Mair P., De Leeuw J. (2015). Unidimensional scaling. In Wiley StatsRef: Statistics Reference Online, Wiley, New York.

**See Also**

[mds](#)

**Examples**

```
## unidimensional scaling of Plato's 7 works
PlatoD <- dist(t(Plato7))
fit.uni <- uniscale(PlatoD)
fit.uni
plot(fit.uni)
```

---

`winedat`*Wine tasting*

---

**Description**

This dataset collects dissimilarity matrices of 10 raters of 6 different wines.

**Usage**

```
data(winedat)
```

**Format**

A list of dissimilarity matrices reflecting the rating of 10 judges on 6 different wines (Ziniel Chardonnay, Markowitsch Chardonnay, Krems Chardonnay, Castel Nova Chardonnay, Ritinitis Noble Retsina, RetsinaCriteria). The attributes color, smell, taste, fun, and overall impression were rated on a scale from 1 (very good) to 5. Based on these ratings the distances were computed.

**Examples**

```
data(winedat)
```

---

`wish_ger`*Wish dataset*

---

**Description**

Similarity ratings for 12 countries. There were no instructions concerning the characteristics on which these similarity judgements were to be made, this was information to discover rather than to impose.

**Usage**

```
data(wish)
```

**Format**

Object of class `dist`

**Details**

Note that for `wish_ger` the country labels are given in German. For `smacof`, the data must be converted into a dissimilarity matrix (see examples).

**References**

Borg, I., Groenen, P. J. F., & Mair, P. (2010). *Multidimensionale Skalierung*. Muenchen: Hampp Verlag.

Borg, I., Groenen, P. J. F., & Mair, P. (2012). *Multidimensional Scaling*. New York: Springer, forthcoming.

Wish, M. (1971). Individual differences in perceptions and preferences among nations. In C. W. King and D. Tigert (Eds.), *Attitude research reaches new heights*, pp. 312-328. Chicago: American Marketing Association.

**Examples**

```
data(wish)
sim2diss(wish)
```



# Index

## \*Topic **datasets**

- bread, 3
- breakfast, 3
- csrranking, 4
- Duration, 5
- ekman, 6
- EW\_ger, 6
- GOPdtm, 7
- Guerry, 9
- helm, 9
- intelligence, 10
- kinshipdelta, 14
- LawLer, 15
- morse, 15
- morsescales, 16
- OCP, 17
- partypref, 17
- perception, 18
- Plato7, 20
- PVQ40, 26
- rectangles, 28
- RockHard, 29
- stardist, 42
- trading, 44
- winedat, 47
- wish\_ger, 47

## \*Topic **hplot**

- jackknife, 12
- plot.smacof, 21

## \*Topic **methods**

- residuals.smacof, 28
- summary.smacofB, 43

## \*Topic **models**

- gravity, 8
- inverseMDS, 11
- permtest, 19
- Procrustes, 24
- randomstress, 27
- sim2diss, 30

- smacofConstraint, 31
- smacofIndDiff, 34
- smacofRect, 36
- smacofSphere, 38
- smacofSym, 40
- torgerson, 44
- uniscale, 46

- bread, 3
- breakfast, 3

- csrranking, 4

- Duration, 5
- DurationRaw (Duration), 5

- ekman, 6
- EW\_eng (EW\_ger), 6
- EW\_ger, 6

- GOPdtm, 7
- gravity, 8
- Guerry, 9

- helm, 9

- idioscal (smacofIndDiff), 34
- indscal (smacofIndDiff), 34
- intelligence, 10
- inverseMDS, 11

- jackknife, 12
- jackknife.smacofB, 20

- kinshipdelta, 14
- kinshipscales (kinshipdelta), 14

- LawLer, 15
- Lawler (LawLer), 15

- mds, 11, 46
- mds (smacofSym), 40

- morse, 15
- morsescapes, 16
- OCP, 17
- partypref, 17
- perception, 18
- permtest, 19
- Plato7, 20
- plot, 13, 22
- plot.procr, 23
- plot.procr (Procrustes), 24
- plot.smacof, 13, 20, 21, 41, 42
- plot.smacofID (plot.smacof), 21
- plot.smacofJK (jackknife), 12
- plot.smacofPerm (permtest), 19
- plot.smacofR (plot.smacof), 21
- plot.uniscale (uniscale), 46
- print.smacofB (summary.smacofB), 43
- print.smacofID (summary.smacofB), 43
- print.smacofJK (jackknife), 12
- print.smacofPerm (permtest), 19
- print.smacofR (summary.smacofB), 43
- print.uniscale (uniscale), 46
- Procrustes, 24
- PVQ40, 26
  
- rainbow\_hcl, 12
- randomstress, 27
- rect\_constr (rectangles), 28
- rectangles, 28
- residuals.smacof, 28
- residuals.smacofID (residuals.smacof), 28
- residuals.smacofR (residuals.smacof), 28
- RockHard, 29
  
- sim2diss, 30
- smacofConstraint, 13, 31, 36, 38, 40, 42
- smacofIndDiff, 33, 34, 38, 40, 42
- smacofRect, 33, 36, 36, 40, 42
- smacofSphere, 33, 36, 38, 38, 42
- smacofSym, 8, 13, 20, 33, 36, 38, 40, 40
- stardist, 42
- summary.smacofB, 43
- summary.smacofID (summary.smacofB), 43
- summary.smacofR (summary.smacofB), 43
  
- torgerson, 44
  
- trading, 44
  
- unfolding (smacofRect), 36
- uniscale, 46
  
- winedat, 47
- wish (wish\_ger), 47
- wish\_ger, 47