Package 'mokken'

July 3, 2016

Version 2.8.3

Date 2016-07-29

Title Perform Mokken Scale Analysis in R

Author L. Andries van der Ark <L.A.vanderArk@uva.nl>

Maintainer L. Andries van der Ark <L.A.vanderArk@uva.nl>

Depends R (>= 2.8.1), graphics, poLCA

Suggests MASS

Description Contains functions for performing Mokken scale analysis on test and questionnaire data. It includes an automated item selection algorithm, and various checks of model assumptions.

License GPL (>= 2)

URL https://sites.google.com/a/tilburguniversity.edu/avdrark/mokken

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-07-03 22:19:20

R topics documented:

mokken-package 2
acl 5
aisp 7
cavalini 8
check.bounds 10
check.ca 11
check.errors 13
check.iio 14
check.monotonicity 16
check.norms 18
check.pmatrix 19
check.reliability 21
check.restscore 22

coefH . . . 24
coefZ . . . 26
DS14 . . . 27
plot.iio.class . . . 28
plot.monotonicity.class . . . 30
plot.pmatrix.class . . . 32
plot.restscore.class . . . 33
recode . . . 35
summary.iio.class . . . 36
summary.monotonicity.class . . . 38
summary.pmatrix.class . . . 39
summary.restscore.class . . . 40
transreas . . . 41
twoway . . . 43

Index **45**

`mokken-package` *Mokken Scale Analysis in R*

Description

Mokken scale analysis (Mokken, 1971; Sijtsma and Molenaar, 2002) is a scaling procedure for both dichotomous and polytomous items. It consists of an item selection algorithm to partition a set of items into Mokken scales and several methods to check the assumptions of two nonparametric item response theory models: the monotone homogeneity model and the double monotonicity model. The output of this R-package resembles the output of the stand-alone program MSP (Molenaar and Sijtsma, 2000).

Details

Package:	mokken
Type:	Package
Version:	2.8.3
Date:	2016-06-29
License:	GPL Version 2 or later

The package contains principal functions for Mokken scale analysis.

The package contains the following data sets

`acl`	Scores on a personality checklist.
`cavalini`	Scores on an inventory on industrial malodor
`transreas`	Scores on a transitive reasoning test
`DS14`	Scores on a Type D test (bootstrap sample)

A guide for Mokken scale analysis in R for people who do not know R (Van der Ark, 2010) is available as a vignette from https://sites.google.com/a/tilburguniversity.edu/avdrark/mokken.

Thanks are due to Geert H. van Kollenburg, Letty Koopman, Renske E. Kuijpers, Rudy Ligtvoet, Hannah E. M. Oosterhuis, J. Hendrik Straat, and Daniel W. van der Palm for contributing R code; to Geert H. van Kollenburg, Letty Koopman, Rudy Ligtvoet, Patrick Mair, and J. Hendrik Straat for testing the software; to Wijbrandt van Schuur for comments on the vignette; to Stephen Cubbellotti, Michael Dewey, Jasmin Durstin, Wilco H. M. Emons, Jue Huang, Michael Kubovy, Ivo Molenaar, Jonathan Rose, Tobias Schlaffer, Klaas Sijtsma, Iris Smits, Jia Jia Syu, Roger Watson, and Na Yang for reporting comments or bugs; to Diederick Stoel (ProfitWise) for financial support, to Harrie C. M. Vorst, Pierre Cavalini, and Johan Denollet for permission to use their data; to Robert J. Mokken for lending his last name.

Version 0 was introduced in Van der Ark (2007). It included the functions

`coefH`	Scalability coefficients
`coefZ`	Test statistics for scalability coefficients
`check.monotonicity`	Investigate monotonicity assumptions
`check.restscore`	Investigate nonintersection assumption using Method Restscore
`check.pmatrix`	Investigate nonintersection assumption using Method Pmatrix
`search.normal`	Mokken's automated item selection algorithm

The following major modifications have been made.

`aisp`	More general automated item selection algorithm.
	Function `search` has become obsolete (Version 2.0)
`check.reliability`	Compute reliability coefficients (Version 2.0)
`check.iio`	Investigate invariant item orderings (Version 2.4)
`coefH`	Standard errors for scalability coefficients included (Version 2.6)

All updates until version 2.7 are described in Van der Ark (2012). The following modifications have been made in Version 2.7 in comparison to previous versions.

`check.errors`	Inclusion new function to compute weighted Guttman errors for each person.
`check.iio`	`plot` has been added.
`check.monotonicity`	Computation of number of active pairs for dichotomous items has been corrected.
`check.pmatrix`	Summary of the results has been corrected.
`check.restscore`	Code pertaining to IIO has been deleted. The procedure is now equivalent to MSP.
`coefH`	Option included to compare scalability coefficients across groups

The following modifications have been made in Version 2.7.1 in comparison to previous versions.

`mokken`	Some legal issues

The following modifications have been made in Version 2.7.2 in comparison to previous versions.

check.iio	Violations of IIO for dichotomous items are now tested using a z-test rather than a t-test.

The following modifications have been made in Version 2.7.3 in comparison to previous versions.

plot.iio.class	Confidence envelopes around estimated response functions
plot.monotonicity.class	Confidence envelopes around estimated response functions
plot.restscore.class	Confidence envelopes around estimated response functions

The following modifications have been made in Version 2.8.1 in comparison to previous versions.

aisp	Startsets have been added

The following modifications have been made in Version 2.8.2 in comparison to previous versions.

recode	New
check.ca	New
check.norms	New
check.errors	Outlier score O+ has been included

The following modifications have been made in Version 2.8.3 in comparison to previous versions.

twoway	New
DS14	New data set
check.errors	Outlier cutoff scores have been included

Author(s)

L. Andries van der Ark Maintainer: L. Andries van der Ark <L.A.vanderArk@uva.nl>.

References

Mokken, R. J. (1971) *A Theory and Procedure of Scale Analysis*. Berlin, Germany: De Gruyter.

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Sijtsma, K., and Van der Ark, L. A. (2016). A tutorial on how to do a Mokken scale analysis on your test and questionnaire data. Manuscript submitted for publication.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software, 20*(11), 1-19. http://www.jstatsoft.org/v20/i11

Van der Ark, L. A. (2010). Getting started with Mokken scale analysis in **R**. Unpublished manuscript. https://sites.google.com/a/tilburguniversity.edu/avdrark/mokken

Van der Ark, L. A. (2012). New developments in Mokken scale analysis in **R**. *Journal of Statistical Software, 48*(5), 1-27. http://www.jstatsoft.org/v48/i5

Examples

```
# Personality test
data(acl)

# Select the items of the scale Communality
Communality <- acl[,1:10]

# Compute scalability coefficients
coefH(Communality)

# Investigate the assumption of monotonicity
monotonicity.list <- check.monotonicity(Communality)
summary(monotonicity.list)
plot(monotonicity.list)

# Investigate the assumption of non-intersecting ISRFs using method restscore
restscore.list <- check.restscore(Communality)
summary(restscore.list)
plot(restscore.list)

# Investigate the assumption of non-intersecting ISRFs using method pmatrix
pmatrix.list <- check.pmatrix(Communality)
summary(pmatrix.list)
plot(pmatrix.list)

# Investigate the assumption of IIO using method MIIO
iio.list <- check.iio(Communality)
summary(iio.list)
plot(iio.list)

# Compute the reliability of the scale
check.reliability(Communality)

# Partition the the scale into mokken scales
aisp(Communality)
```

`acl` *Adjective Checklist Data*

Description

Scores of 433 students on 218 items from a Dutch version of the Adjective Checklist.

Usage

```
data(acl)
```

Format

A 433 by 218 matrix containing integers. `dimnames(acl)[[2]]` are adjectives

Details

Each item is an adjective with five ordered answer categories (0 = completely disagree, 1 = disagree, 2 = agree nor disagree, 3 = agree, 4 = completely agree). The respondents were instructed to consider whether an adjective described their personality, and mark the answer category that fits best to this description. The 218 items constitute 22 scales (see table); 77 items of the 218 items that constitute the ten scales were negatively worded. The negatively worded items are indicated by an asterisk in the `dimnames` and their item scores have been reversed. The Deference scale measures in fact the opposite of Deference.

Communality	Items 1-10	Change	Items 111-119
Achievement	Items 11-20	Succorance	Items 120-129
Dominance	Items 21-30	Abasement	Items 130-139
Endurance	Items 31-40	Deference*	Items 140-149
Order	Items 41-50	Personal Adjustment	Items 150-159
Intraception	Items 51-60	Ideal Self	Items 160-169
Nurturance	Items 61-70	Critical parent	Items 170-179
Affiliation	Items 71-80	Nurturant parent	Items 180-189
Exhibition	Items 81-90	Adult	Items 190-199
Autonomy	Items 91-100	Free Child	Items 200-209
Aggression	Items 101-110	Adapted Child	Items 210-218

Source

Data were kindly made available by H. C. M. Vorst from the University of Amsterdam. The original Adjective Checklist was developed by Gough and Heilbrun (1980).

References

Gough, H. G. and Heilbrun,A. B. (1980) *The Adjective Check List, Manual 1980 Edition*. Palo Alto, CA: Consulting Psychologists Press.

Van der Ark, L. A. (2007) Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

Examples

```
data(acl)
```

`aisp` *Automated Item Selection Procedure (AISP) for Mokken Scale Analysis*

Description

Returns a vector with as many elements as there are items, indicating to which scale an item belongs.

Usage

```
aisp(X, search="normal", lowerbound=.3, alpha=.05, StartSet=FALSE, popsize=20,
maxgens=default.maxgens, pxover=0.5, pmutation=0.1, verbose=FALSE)
```

Arguments

`X`	matrix or data frame of numeric data containing the responses of `nrow(X)` respondents to `ncol(X)` items. Missing values are not allowed
`search`	Type of item selection procedure: "normal": Mokken's automated item selection procedure (Mokken, 1971; Molenaar and Sijtsma, 2000; Sijtsma and Molenaar, 2002); "ga": item selection using a genetic algorithm (Straat, van der Ark, and Sijtsma, 2013). The default is "normal".
`lowerbound`	numeric scaling criterium; 0 <= `lowerbound` < 1. The default is `0.3`.
`alpha`	Type I error level. The default is `0.05`.
`StartSet`	Startset of items for the first scale. Vector of item numbers. If `StartSet == FALSE` no startset is provided (default).
`popsize`	Size of the population of items in genetic. algorithm The default is `20`.
`maxgens`	Number of generations in genetic algorithm. The default is `10^(log2(ncol(X)/5)) * 1000`.
`pxover`	Cross-over probability in genetic algorithm. The default is `0.5`.
`pmutation`	Mutation probability in genetic algorithm. The default is `0.1`.
`verbose`	Logical, indicating whether should output to the screen the results of the model. If `FALSE`, no output is produced. The default is `TRUE`.

Details

Each scale must consist of at least two items, hence the number of Mokken scales cannot exceed `ncol(X)/2`. Procedure may be slow for large data sets. Especially if the genetic algorithm is used. There is not yet an option `search="extended"`. `aisp` replaces the function `search.normal` in earlier versions.

Value

An indicator vector of length J. Each entry refers to an item. Items with same integer belong to the same Mokken scale. A zero indicates an unscalable item. If *n* is the largest integer, then *n* Mokken scales were found.

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>, J. H. Straat

References

Mokken, R. J. (1971) *A Theory and Procedure of Scale Analysis*. Berlin, Germany: De Gruyter.

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Straat, J. H., Van der Ark, L. A., and Sijtsma, K. (2013). Comparing optimization algorithms for item selection in Mokken scale analysis. *Journal of Classification, 30,* 72-99.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

See Also

coefH, check.iio, check.monotonicity, check.pmatrix, check.reliability,check.restscore

Examples

```
data(acl)

# Select the scale Communality consisting of 10 items.
Communality <- acl[,1:10]

# Partition these 10 items into mokken scales using Mokken's automated item selection procedure.
scale <- aisp(Communality)
coefH(Communality[,scale==1])

# Same but using items 1 and 2 in the startset.
scale <- aisp(Communality, StartSet = c(1, 2), verbose = TRUE)
coefH(Communality[,scale==1])

# Partition these 10 items into mokken scales using a genetic algorithm.
scale <- aisp(Communality,"ga",maxgens=1000)
coefH(Communality[,scale==1])
```

---

cavalini *Coping Strategies*

Description

Data came from 17 polytomous items administered to 828 respondents (Cavalini, 1992) asking them how they coped actively with the bad smell from a factory in the neighborhood of their homes.

Usage

```
data(cavalini)
```

Format

A 828 by 17 matrix containing integers. `attributes(transreas)` gives details on the items.

Details

Items have four ordered answer categories, *never* (score 0), *seldom* (1), *often* (2), and *always* (3). The 17 items constitute 4 scales (for detailed information, see Sijtsma and Molenaar, 2002, pp. 82-86).

Item1	Keep windows closed
Item2	No laundry outside
Item3	Search source of malodor
Item4	No blankets outside
Item5	Try to find solutions
Item6	Go elsewhere for fresh air
Item7	Call environmental agency
Item8	Think of something else
Item9	File complaint with producer
Item10	Acquiesce in odor annoyance
Item11	Do something to get rid of it
Item12	Say "it might have been worse"
Item13	Experience unrest
Item14	Talk to friends and family
Item15	Seek diversion
Item16	Avoid breathing through the nose
Item17	Try to adapt to situation

References

Cavalini, P. M. (1992). *It's an ill wind that brings no good. Studies on odour annoyance and the dispersion of odorant concentrations from industries.* Unpublished doctoral disseratation. University of Groningen, The Netherlands.

Sijtsma, K. and Molenaar, I. W. (2002). *Introduction to nonparametric item response theory*, Thousand Oaks, CA: Sage.

See Also

`check.iio`,

Examples

```
data(cavalini)
attributes(cavalini)$labels
```

check.bounds *Check the relative lower bound for scalability coefficients*

Description

Returns the relative bounds for Mokken's scalability coefficients for dichotomous items as described by Ellis (2014).

Usage

```
check.bounds(X, quant = .90, lower = TRUE, upper = FALSE)
```

Arguments

X	matrix or data frame of numeric data containing the responses of nrow(X) respondents to ncol(X) items. Missing values are not allowed
quant	numerical value between 0 and 1 used for the computation of lower bound L2rij. The computation deviates somewhat from the proposal in Ellis (2014) because the stats function quantile is used.
lower	Boolean: If TRUE, the lower bounds are given.
upper	Boolean: If TRUE, the upper bounds are given.

Value

List containing two lists UpperBounds and LowerBounds, each containing a list of two J x J matrices (J = number of items): L1rij (overestimator of the lower bound for the correlation) and L2rij (underestimator of the lower bound for the correlation).

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Ellis, J. L. (2014). An inequality for correlations in unidimensional monotone latent variable models for binary variables. *Psychometrika, 79*, 303-316.

Examples

```
data(acl)
Communality <- acl[,1:10]
R <- cor(Communality)
res <- check.bounds(Communality, upper = TRUE)
L1rij <- res$LowerBounds$L1rij
L2rij <- res$LowerBounds$L2rij
U1rij <- res$UpperBounds$U1rij
U2rij <- res$UpperBounds$U2rij
# Correlations that meet L1rij (possibly an overestimation of the lower bound).
```

```
R >= L1rij
# Correlations that meet U1rij (possibly an overestimation of the upper bound).
R <= U1rij
# Correlations that meet L2rij (possibly an underestimation of the lower bound).
R >= L2rij
# Correlations that meet U2rij (possibly an underestimation of the upper bound).
R <= U2rij
```

---

check.ca *Check conditional association to indentify local dependence.*

Description

The function uses three special cases of conditional association (CA; Holland and Rosenbaum, 1986) to identify positive and negative local dependence in the monotone homogeneity model. Straat, Van der Ark, and Sijtsma (2016; also, see Sijtsma, Van der Ark, and Straat, 2015) described the procedure.

Usage

```
check.ca(X, Windex = FALSE, MINSIZE = 4, NWEIGHTOPTION = "noweight",
            COVWEIGHTOPTION = "pnorm", MINGROUP = 4)
```

Arguments

X	Matrix of integers, missing values are not allowed
Windex	Boolean. Should output contain indices W1, W2, and W3?
MINSIZE	Minimum sample size of a rest-score group
NWEIGHTOPTION	Weight of sample size on each conditional covariance. Options: "noweight" (each covariance has weight 1, default in Straat et al., 2016) and "sqrt" (each covariance has weight sqrt(N_k(x)), this option was used in an older, decrepit, version of Straat et al., 2016)
COVWEIGHTOPTION	Weight of each conditional covariance on the computation of W1, W2, and W3. Options: "pnorm" (weight equals P[cov < 0], default in Straat et al., 2014) and "noweight" (if cov < 0, then weight = 1, and weight = 0, otherwise; this option was used in a previous version of Straat et al., 2014)
MINGROUP	Minimum sample size of the conditioning variable to compute a covariance. Since the term N-3 is used in the computation of the standard error, N = 4 is the default.

Value

list of three components:

(1) `InScale` (vector of booleans with length equal to the number of items): indicates whether an item is still in the scale.

(2) `Index` (list): Numerical values of indices W1, W2, and W3 (shown only if `Windex = TRUE`). `Index` has three subcomponents: `W1`, `W2`, and `W3`.

(3) `Flagged` (list): Boolean indicating whether a value of W1, W2, and W3 is flagged (1) or not (0) (shown only if `Windex = TRUE`) `Index` has three subcomponents: `F1`, `F2`, and `F3`, corresponding to the flagging of indices W1, W2, and W3, respectively.

Subcomponents correspond to the iteration. The first subcomponent refers to the situation with all items in the test, the second subcomponent refers to the sitution with the worst item deleted, the third subcomponent refers to the sitution with the two worst items deleted, etc.

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl> and J. H. Straat

References

Straat, J. H., van der Ark, L. A., and Sijtsma, K. (2016). Using conditional association to identify local dependence. Manuscript submitted for publication.

Sijtsma, K., van der Ark, L. A., and Straat, J. H. (2015) Goodness of fit methods for nonparametric IRT models. In L. A. van der Ark, D. M. Bolt, W.-C. Wang, J. Douglas, and S.-M. Chow (Eds.), *Quantitative psychology research: The 79th Annual Meeting of the Psychometric Society, Madison, Wisconsin, 2014.* (pp. 109 - 120) New York, NY: Springer.

See Also

DS14, recode, twoway

Examples

```
data(DS14)

# Handle missing data and recode negatively worded items
X <- DS14[, 3 : 16]
X <- twoway(X)
X <- recode(X, c(1, 3))

# Negative affectivity
Na <- X[, c(1, 3, 6, 8, 10, 11, 14)]

# Social inhibition
Si <- X[, c(2, 4, 5, 7, 9, 12, 13)]

check.ca(Na, TRUE)
```

`check.errors` *Check the number of Guttman errors (Gplus) and the number of infrequent scores (Oplus) for each respondent*

Description

Returns a lost containing outlier scores Gplus (number of Guttman errors; Guttman, 1944) and Oplus for each respondent (Zijlstra, van der Ark and Sijtsma, 2007).

Usage

```
check.errors(X, returnGplus = TRUE, returnOplus = FALSE)
```

Arguments

`X`	matrix or data frame of numeric data containing the responses of `nrow(X)` respondents to `ncol(X)` items. Missing values are not allowed
`returnGplus`	Boolean. If `TRUE` the output contains outlier score Gplus
`returnOplus`	Boolean. If `TRUE` the output contains outlier score Oplus

Value

List. Depending on the values of `returnGplus` and `returnOplus`, the output contains outlier score Gplus (the number of Guttman errors) and Oplusfor each respondent

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Guttman, L. (1944) A basis for scaling qualitative data. *American Sociological Review, 9*, 139-150.

Meijer, R. R. (1994) The number of Guttman errors as a simple and powerful person-fit statistic. *Applied Psychological Measurement, 18*, 311-314.

Mokken, R. J. (1971) *A Theory and Procedure of Scale Analysis*. Berlin, Germany: De Gruyter.

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

Zijlstra, W. P., Van der Ark, L. A., and Sijtsma, K. (2007). Outlier detection in test and questionnaire data. *Multivariate Behavioral Research, 42*, 531-555.

See Also

check.ca, check.iio, check.monotonicity, check.pmatrix, check.reliability coefH, plot.restscore.class, summary.restscore.class

Examples

```
data(acl)
Communality <- acl[,1:10]
Gplus <- check.errors(Communality, TRUE, FALSE)$Gplus
Oplus <- check.errors(Communality, FALSE, TRUE)$Oplus

hist(Gplus, breaks = 0:max(Gplus))
```

---

check.iio *Check of Invariant Item Ordering*

Description

Returns a list (of class `iio.class`) with results from the investigation of invariant item ordering. Three methods may be used for the investigation of invariant item ordering. (1) Method MIIO (manifest invariant item ordering: investigates the manifest item response functions for all pairs of items). For polytomous items, t-tests are used to test violations, for dichotomous items z-tests are used to test are used to test violations. (2) Method MS-CPM (manifest scale - cumulative probability model: investigates the manifest item step response functions for all pairs of items). Z-tests are used to test violations. (3) Method IT (increasingness in transposition: investigates all bivariate joint probabilities for all pairs of items). Chi-square tests are used to test violations.

For a complete description of Method MIIO, see Ligtvoet, Van der Ark, Te Marvelde, and Sijtsma (2010); for a complete description of the Method MS-CPM and Method IT with reference to Method MIIO, see Ligtvoet, Van der Ark, Bergsma, and Sijtsma (2011).

Usage

```
check.iio(X, method="MIIO", minvi = default.minvi, minsize = default.minsize,
alpha = .05, item.selection=TRUE, verbose=FALSE)
```

Arguments

`X`	matrix or data frame of numeric data containing the responses of `nrow(X)` respondents to `ncol(X)` items. Missing values are not allowed
`method`	Either `"MIIO"` (default), `"MSCPM"`, or `"IT"`. Partial matching is allowed (e.g. `method="ms"` is equivalent to `method="MSCPM"`)
`minvi`	minimum size of a violation that is reported. By default `minvi` = .03 times the number of item step response functions (m) for Method MIIO; `minvi` = .03 for Method IT and Method MSCPM
`minsize`	minimum size of a rest score group. By default `minsize` = $N/10$ if $N \geq 500$; `minsize` = $N/5 if\ 250 \leq N < 500$; and `minsize` = $\max(N/3, 50)$ if $N < 250$

alpha	Nominal Type I error for t test (Method MIIO), z test (Method MSCPM), or McNemar test (Method IT). Default alpha = .05
item.selection	Conduct backward item selection procedure (see Ligtvoet et al., 2010). Default item.selection=TRUE
verbose	Show the results of the backward item selection algorithm on screen. Default verbose=FALSE

Details

The output is of class iio.class, and is often numerous. Functions plot and summary can be used to summarize the output. See Van der Ark (2014) for an example.

Value

results	A list with as many components as there are item pairs. Each component itself is also a list containing the results of the investigation of IIO.
violations	A matrix: Summary of the backward item selection (Corresponds to Table 4 in Ligtvoet et al., 2010, and Table 1 in Ligtvoet et al., 2011). The first column gives, for each item, the number of violations of IIO. If the number of violations is nonzero, then the item with the largest number of violations is removed. If two or more items have the maximum number of violations, then from those items the item producing the lowest value of Loevinger's H is removed. The second column shows the number of violations with one item removed, the third column shows the number of violations with two items removed, etc.
items.removed	List of the items removed in chronological order
HT	Coefficient HT for the remaining items. For the use of coefficient HT see Ligtvoet et al. (2010). If the sample size is extremely large coefficient HT is estimated using a random subsample.
method	The argument method
item.mean	The mean item scores

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Ligtvoet, R., L. A. van der Ark, J. M. te Marvelde, and K. Sijtsma (2010). Investigating an invariant item ordering for polytomously scored items. *Educational and Psychological Measurement, 70*, 578-595.

Ligtvoet, R., L. A. van der Ark, W. P. Bergsma, and K. Sijtsma (2011). Polytomous latent scales for the investigation of the ordering of items. *Psychometrika, 76*, 200-216.

Sijtsma, K., R. R. Meijer, and L. A. van der Ark (2011). Mokken scale analysis as time goes by: An update for scaling practitioners. *Personality and Individual Differences, 50*, 31-37.

Sijtsma, K, and I. W. Molenaar (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software, 20* (11), 1-19. http://www.jstatsoft.org/v20/i11

Van der Ark, L. A. (2012). New developments in Mokken scale analysis in R. *Journal of Statistical Software, 48*(5), 1-27. http://www.jstatsoft.org/v48/i5

See Also

check.errors, check.monotonicity, check.pmatrix, check.reliability check.restscore, coefH, plot.iio.class, summary.iio.class,

Examples

```
# Examples from Ligtvoet et al. (2010).

data(acl)

Order <- acl[,41:50]
summary(check.iio(Order))
plot(check.iio(Order))

Autonomy <- acl[,91:100]
check.iio(Autonomy)

# Examples from Ligtvoet et al. (2011).

data(cavalini)
X1 <- cavalini[,c(3,5,6,7,9,11,13,14)]

# Use Method MIIO and remove items violating MIIO
iio.list1 <- check.iio(X1)
summary(iio.list1)
X2 <- X1[,is.na(charmatch(dimnames(X1)[[2]],names(iio.list1$items.removed)))]

# Use Method MSCPM and remove items violating MSCPM
iio.list2 <- check.iio(X2,method="MSCPM")
summary(iio.list2)
X3 <- X2[,is.na(charmatch(dimnames(X2)[[2]],names(iio.list2$items.removed)))]

# Use Method IT
iio.list3 <- check.iio(X3,method="IT")
summary(iio.list3)
```

---

check.monotonicity *Check of Monotonicity*

Description

Returns a list (of class `monotonicity.class`) with results from the investigation of monotonicity (Junker and Sijtsma, 2000; Mokken, 1971; Molenaar and Sijtsma, 2000; Sijtsma and Molenaar, 2002).

Usage

```
check.monotonicity(X, minvi = 0.03, minsize = default.minsize)
```

Arguments

`X`	matrix or data frame of numeric data containing the responses of `nrow(X)` respondents to `ncol(X)` items. Missing values are not allowed
`minvi`	minimum size of a violation that is reported
`minsize`	minimum size of a rest score group. By default `minsize` = $N/10$ if $N \geq 500$; `minsize` = $N/5 if\ 250 \leq N < 500$; and `minsize` = $\max(N/3, 50)$ if $N < 250$

Details

The output is of class monotonicity.class, and is often numerous. Functions `plot` and `summary` can be used to summarize the output. See Van der Ark (2007) for an example.

Value

`results`	A list with as many components as there are items. Each component itself is also a list containing the results of the check of monotonicity.
`I.labels`	The item labels
`Hi`	The item scalability coefficients Hi
`m`	The number of answer categories.

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Junker, B.W., and Sijtsma, K. (2000). Latent and manifest monotonicity in item response models. *Applied Psychological Measurement, 24*, 65-81.

Mokken, R. J. (1971) *A Theory and Procedure of Scale Analysis*. Berlin, Germany: De Gruyter.

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

See Also

check.errors, check.iio, check.restscore, check.pmatrix, check.reliability, coefH, plot.monotonicity.class
summary.monotonicity.class

Examples

```
data(acl)
Communality <- acl[,1:10]
monotonicity.list <- check.monotonicity(Communality)
plot(monotonicity.list)
summary(monotonicity.list)
```

---

check.norms *Standard errors for norm statistics*

Description

The function presents standard errors for the mean, standard deviation, standard scores, stanine boundaries, and percentiles based on a vector of test scores (Oosterhuis, Van der Ark, and Sijtsma, 2016).

Usage

```
check.norms(y, nice.output = TRUE)
```

Arguments

y	numerical vector. Typically a numerical vector of length N, representing the test scores of N respondents. Missing values are not allowed
nice.output	Logical: If TRUE, norm statistics and standard errors are combined in an a single object of class noquote

Value

list of five components:

(1) mean: Sample mean and its standard error (noquote).

(2) sd: Sample standard deviation and its standard error (noquote).

(3) z: For each unique testscore, the test score, its frequency, the corresponding estimated standard score and its standard error (noquote).

(4) sta9: The estimates of the 8 boundaries of the stanines and their standard error (noquote).

(5) z: For each unique testscore, the test score, its frequency, the corresponding estimated percentile rank and its standard error (noquote).

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl> and H. E. M. Oosterhuis

References

Oosterhuis, H. E. M., van der Ark, L. A., and Sijtsma, K. (2016). Standard errors and confidence intervals of norm statistics for educational and psychological tests. Manuscript submitted for publication.

See Also

DS14, recode, twoway

Examples

```
data(DS14)

# Handle missing data and recode negatively worded items
X <- DS14[, 3 : 16]
X <- twoway(X)
X <- recode(X, c(1, 3))

# Negative affectivity
Na <- X[, c(1, 3, 6, 8, 10, 11, 14)]

# Social inhibition
Si <- X[, c(2, 4, 5, 7, 9, 12, 13)]

# Norms
check.norms(rowSums(Na))
check.norms(rowSums(Si))
```

---

check.pmatrix *Check of Nonintersection Using Method Pmatrix*

Description

Returns a list (of class `pmatrix.class`) with results from the investigation of nonintersection using method pmatrix (Mokken, 1971; Molenaar and Sijtsma, 2000; Sijtsma and Molenaar, 2002).

Usage

```
check.pmatrix(X, minvi = 0.03)
```

Arguments

`X`	matrix or data frame of numeric data containing the responses of `nrow(X)` respondents to `ncol(X)` items. Missing values are not allowed
`minvi`	minimum size of a violation that is reported

Details

The output is often numerous. Functions `plot` and `summary` can be used to summarize the output. See Van der Ark (2007) for an example.

Value

`results`	A list with as many components as there are item pairs. Each component itself is also a list containing the results of the check of nonintersection using Method pmatrix. The P(++) matrix and P(- -) (Molenaar and Sijtsma, 2000; Sijtsma and Molenaar, 2002) are also included.
`I.item`	vector indicating to which items the rows and column the P(++) matrix belong
`I.step`	the labels of the item steps in order of popularity
`I.labels`	the item labels
`Hi`	the item scalability coefficients Hi
`minvi`	the value of minvi

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Mokken, R. J. (1971) *A Theory and Procedure of Scale Analysis*. Berlin, Germany: De Gruyter.

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

See Also

`check.errors`, `check.iio`, `check.monotonicity`, `check.reliability` `check.restscore`, `coefH`, `plot.pmatrix.class`, `summary.pmatrix.class`

Examples

```
data(acl)
Communality <- acl[,1:10]
pmatrix.list <- check.pmatrix(Communality)
plot(pmatrix.list)
summary(pmatrix.list)
```

check.reliability *Computation of reliability statistics*

Description

Returns a list of reliability statistics: Molenaar Sijtsma (MS, 1984, 1988) statistic (a.k.a rho; also see, Sijtsma and Molenaar, 1987; Van der Ark, 2010), Cronbach's (1951) alpha, Guttman's (1945) lambda 2, and the latent class reliability coefficient (LCRC; Van der Ark, Van der Palm, and Sijtsma, 2011).

Usage

```
check.reliability(X, MS = TRUE, alpha = TRUE, lambda.2 = TRUE,
LCRC = FALSE, nclass = nclass.default, irc = FALSE)
```

Arguments

X	matrix or data frame of numeric data containing the responses of nrow(X) respondents to ncol(X) items. Missing values are not allowed
MS	Boolean. If TRUE, The MS statistic is computed.
alpha	Boolean. If TRUE, Cronbach's alpha is computed.
lambda.2	Boolean. If TRUE, Guttman's Lambda 2 is computed.
LCRC	Boolean. If TRUE, the LCRC is computed.
nclass	Integer. Number of latent classes for the computation of LCRC. By default: the number of items minus 1.
irc	Boolean.If TRUE, the item-rest correlation (a.k.a. corrected item-total correlation) is computed.

Details

The computation of LCRC depends on the package poLCA, which in its turn depends on the packages MASS and scatterplot3d. Computation of the LCRC may be time consuming if the number of latent classes is large. The optimal number of latent classes should be determined prior to the computation of the LCRC, using software for latent class analysis (e.g., the R-package poLCA).

Value

MS	Molenaar Sijtsma statistic (a.k.a. rho).
alpha	Cronbach's alpha
lambda.2	Guttman's Lambda 2
LCRC	LCRC

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Cronbach, L. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika, 16*, 297-334.

Guttman, L. (1945). A basis for analyzing test-retest reliability. *Psychometrika, 10*,255-282.

Molenaar, I. W. and K. Sijtsma (1984). Internal consistency and reliability in Mokken's nonparametric item response model. *Tijdschrift voor onderwijsresearch, 9*, 257–268.

Molenaar, I. W. and K. Sijtsma (1988). Mokken's approach to reliability estimation extended to multicategory items. *Kwantitatieve methoden, 9(28)*, 115-126.

Sijtsma, K. and I. W. Molenaar (1987). Reliability of test scores in nonparametric item response theory. *Psychometrika, 52*,79-97.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

Van der Ark, L. A. (2010). Computation of the Molenaar Sijtsma statistic. In A. Fink, B. Lausen, W. Seidel, and A. Ultsch (Eds.), *Advances in data analysis, data handling and business intelligence* (pp. 775-784). Berlin: Springer.

Van der Ark, L. A., D. W. van der Palm, and K. Sijtsma (2011). A latent class approach to estimating test-score reliability. *Applied Psychological Measurement, 35*, 380-392.

See Also

`check.errors`, `check.iio`, `check.monotonicity`, `check.pmatrix` `check.restscore`, `coefH`

Examples

```
data(acl)
Communality <- acl[,1:10]
check.reliability(Communality, LCRC = TRUE)
```

`check.restscore` *Check of Nonintersection Using Method Restscore*

Description

Returns a list (of class `restscore.class`) with results from the investigation of nonintersection using method restscore (Mokken, 1971; Molenaar and Sijtsma, 2000; Sijtsma and Molenaar, 2002).

Usage

```
check.restscore(X, minvi = 0.03, minsize = default.minsize)
```

Arguments

X	matrix or data frame of numeric data containing the responses of nrow(X) respondents to ncol(X) items. Missing values are not allowed
minvi	minimum size of a violation that is reported
minsize	minimum size of a rest score group. By default minsize = $N/10$ if $N \geq 500$; minsize = $N/5 if\ 250 \leq N < 500$; and minsize = $\max(N/3, 50)$ if $N < 250$

Details

The output is often numerous. Procedure may be slow for large data sets. Functions plot and summary can be used to summarize the output. See Van der Ark (2007) for an example.

Value

results	A list with as many components as there are item pairs. Each component itself is also a list containing the results of the check of nonintersection using method restscore.
I.labels	The item labels
Hi	The item scalability coefficients
m	The number of answer categories.

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Mokken, R. J. (1971) *A Theory and Procedure of Scale Analysis*. Berlin, Germany: De Gruyter.

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

See Also

check.errors, check.iio, check.monotonicity, check.pmatrix, check.reliability coefH, plot.restscore.class, summary.restscore.class

Examples

```
data(acl)
Communality <- acl[,1:10]
restscore.list <- check.restscore(Communality)
plot(restscore.list)
summary(restscore.list)
```

coefH *Scalability coefficents H*

Description

Computes item-pair scalability coefficents Hij, item scalability coefficents Hi, and scale scalability coefficent H (Loevinger, 1948; Mokken, 1971, pp. 148-153; Molenaar and Sijtsma, 2000, pp. 11-13; Sijtsma and Molenaar, chap. 4; Van der Ark, 2007; 2010), as well as their standard errors (Kuijpers, Van der Ark, and Croon, 2013; also see Van der Ark, Croon, and Sijtsma, 2008). It is possible to compare scalability coefficients across groups using the item-step ordering of the entire sample (cf. CHECK=GROUPS option in MSP; Molenaar and Sijtsma, 2000).

Usage

```
coefH(X, se = TRUE, nice.output = TRUE, group.var = NULL)
```

Arguments

X	matrix or data frame of numeric data containing the responses of nrow(X) respondents to ncol(X) items. Missing values are not allowed
se	Logical: If TRUE, the standard errors of the scalability coefficients are given
nice.output	Logical: If TRUE, scalability coefficients and standard errors are combined in an a single object of class noquote
group.var	vector of length nrow(X) or matrix with number of rows equal to nrow(X) to be used as grouping variable

Details

May not work if any of the item variances equals zero. Such items should not be used in a test and removed from the data frame.

If nice.output = TRUE and se = TRUE, the result is a list of 3 objects of class noquote; if nice.output = FALSE and se = TRUE, the result is a list of 6 matrices (3 for the scalability coefficients and 3 for the standard errors); and if se = FALSE, the result is a list of 3 matrices (for the scalability coefficients. if group.var = Y with Y having K values, an additional element named Groups is added to the list. Element Groups shows the scalability coefficients per group ordered by means of sort (see Sys.getlocale for details). group.var returns coefficients for groups containing at least two case. Computation of standard errors can be slow for a combination of a large sample size and a large number of items.

Value

Hij	scalability coefficients of the item pairs (possibly with standard errors; see details)
Hi	vector containing scalability coefficients of the items (possibly with standard errors; see details)

H	scalability coefficient of the entire scale (possibly with standard error; see details)
se.Hij	standard errors of the scalability coefficients of the item pairs (only if nice.output = FALSE and se = TRUE; see details)
se.Hi	standard errors of the scalability coefficients of the items (see details)
se.H	standard error of the scalability coefficient of the entire scale (see details)
Groups	Scalability coefficient for subgroups (see details)

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Kuijpers, R. E., Van der Ark, L. A., and Croon, M. A. (2013). Standard errors and confidence intervals for scalability coefficients in Mokken scale analysis using marginal models. *Sociological Methodology, 43,* 42-69.

Loevinger, J. (1948). The technique of homogeneous tests compared with some aspects of 'scale analysis' and factor analysis. *Psychological Bulletin, 45*, 507-530.

Mokken, R. J. (1971) *A Theory and Procedure of Scale Analysis*. Berlin, Germany: De Gruyter.

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K., and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software, 20* (11), 1-19. http://www.jstatsoft.org/v20/i11

Van der Ark, L. A. (2010). Getting started with Mokken scale analysis in **R**. Unpublished manuscript. https://sites.google.com/a/tilburguniversity.edu/avdrark/mokken

Van der Ark, L. A., Croon, M. A., and Sijtsma (2008). Mokken scale analysis for dichotomous items using marginal models. *Psychometrika, 73*, 183-208.

See Also

coefZ, search.normal

Examples

```
data(acl)
Communality <- acl[,1:10]
coefH(Communality)

coefH(Communality, se=FALSE)

subgroup <- ifelse(acl[,11] < 2,1,2)
coefH(Communality, group.var = subgroup)
```

coefZ *Computation of Z-Values*

Description

Computes Zij-values of item pairs, Zi-values of items, and Z-value of the entire scale, which are used to test whether Hij, Hi, and H, respectively, are significantly greater than zero (Molenaar and Sijtsma, 2000, pp. 59-62; Sijtsma and Molenaar, p. 40; Van der Ark, 2007; 2010).

Used in the function `aisp`

Usage

```
coefZ(X)
```

Arguments

`X`	matrix or data frame of numeric data containing the responses of `nrow(X)` respondents to `ncol(X)` items. Missing values are not allowed

Details

Unlike `coefH`, standard errors are not provided.

Value

`Zij`	matrix containing the Z-values of the item-pairs
`Zi`	vector containing Z-values of the items
`Z`	Z-value of the entire scale

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

Van der Ark, L. A. (2010). Getting started with Mokken scale analysis in **R**. Unpublished manuscript. https://sites.google.com/a/tilburguniversity.edu/avdrark/mokken

See Also

`coefH`, `aisp`

Examples

```
data(acl)
Communality <- acl[,1:10]
coefH(Communality)
coefZ(Communality)
```

DS14 *DS14*

Description

Gender, age, and item scores on the DS14 questionnaire of 541 coronary artery disease patients.

Usage

```
data(DS14)
```

Format

A 541 by 16 matrix containing gender, age, and item scores on the DS14 questionnaire.

Details

The DS14 (Denollet, 2005) is the most accepted and widely used diagnostic instrument for the assessment of the type-D pattern. Type D (distressed) is defined as the joint tendency towards negative affectivity (e.g., worry, irritability, gloom) and social inhibition (e.g., reticence and a lack of self-assurance). DS14 contains 14 items, each having five ordered response categories (0 = completely disagree, 1 = disagree, 2 = agree nor disagree, 3 = agree, 4 = completely agree). Items 2, 4, 5, 7, 9, 12, and 13 measure negative affectivity. Items 1, 3, 6, 8, 10, 11, and 14 measure social inhibition. Items 1 and 3 are negatively worded (indicated by an asterisk in the `dimnames`).

The data contain the gender (`Male`) of the patients (1 = male, 0 = female), the age (`Age`) of the patients in years, and the scores to DS14. Ten item scores are missing. Items 1 and 3 must be recoded before the data can beused meaningfully. The data have been used to investigate predictive value of social inhibition and negative affectivity for cardiovascular events and mortality in patients with coronary artery disease (Denollet et al., 2013), to investigate the relation between Type D and inflammation and endothelial dysfunction' (van Dooren et al., 2016), and to investigate the relation between Type D and increased macrophage activity (Zuccarella-Hackl et al., 2016) These data have also been analyzed in papers on Mokken scale analysis (Sijtsma and Van der Ark, 2016; Straat et al., 2016).

Source

Data were kindly made available by J. Denollet from Tilburg University.

References

Denollet, J., Pedersen, S. S., Vrints, C. J., and Conraads, V. M. (2013). Predictive value of social inhibition and negative affectivity for cardiovascular events and mortality in patients with coronary artery disease: the Type D personality construct. *Psychosomatic Medicine, 75*, 873-981.

van Dooren, F. E., Verhey, F. R., Pouwer, F., Schalkwijk, C. G., Sep, S. J., Stehouwer, C.D., Henry, R. M., Dagnelie, P. C., Schaper, N. C., Van der Kallen, C. J., Koster, A., Schram, M. T., and Denollet, J. (2016). Association of Type D personality with increased vulnerability to depression: Is there a role for inflammation or endothelial dysfunction? - The Maastricht Study. *Journal of Affective Disorders, 189*, 118-125.

Sijtsma, K., and Van der Ark, L. A. (2016). A tutorial on how to do a Mokken scale analysis on your test and questionnaire data. Manuscript submitted for publication.

Straat, J. H., van der Ark, L. A., and Sijtsma, K. (2016). Using conditional association to identify local dependence. Manuscript submitted for publication.

Zuccarella-Hackl, C., von K\"anel, R., Thomas, L., Kuebler, P., Schmid, J. P., Mattle. H. P., Mono, M. L., Rieben, R., Wiest, R., and Wirtz, P. H. (2016). Higher macrophage superoxide anion production in coronary artery disease (CAD) patients with Type D personality. *Psychoneuroendocrinology, 68*, 186-193.

See Also

recode, twoway

Examples

```
data(DS14)

# Handle missing data and recode negatively worded items
X <- DS14[, 3 : 16]
X <- twoway(X)
X <- recode(X, c(1, 3))
head(X)
```

---

plot.iio.class *Plot iio.class objects*

Description

S3 Method to plot objects of class iio.class. Graphic display of the checks of iio. One graph for each item plotting the estimated item response functions.

Usage

```
## S3 method for class 'iio.class'
plot(x, item.pairs = all.pairs, ci = TRUE, alpha = .05,
       color = c("black", "blue"), transparancy = 20, ask = TRUE, ...)
```

Arguments

x	Object of class iio.class produced by check.iio.
item.pairs	vector containing the numbers of the item pairs for which the results are depicted graphically. For example, item.pairs = 1 prints the results for items 1 and 2, item.pairs = 2 prints the results for items 1 and 3, item.pairs = J prints the results for items 1 and J, and item.pairs = J+1 prints the results for items 2 and 3. Default the results for all item pairs are depicted.
ci	Boolean. If TRUE (default), then confidence envelops are plotted around IRFs.
alpha	Type of plotted (1 - alpha) confidence intervals. By default 95-percent confidence intervals are depicted
color	Color of the plotted curves and confidence envelops. Defaults are black for the first item and blue for the second item.
transparancy	Transparancy of the confidence intervals. Higher values result in more opaque colors for the confidence intervals.
ask	Boolean. If TRUE (default), then par("ask"=TRUE); i.e., a hard return between subsequent plots is required. If FALSE, then par("ask"=FALSE).
...	Optional graphical parameters will be ignored

Details

The plot function corresponds to method MIIO; each graph plots the estimated item response functions (item rest-score functions) for two items. For details of the method, see Ligtvoet et al. (2010, 2011); Sijtsma et al. (2012). For details of the confidence envelopes, see Van der Ark (2012b). For the implementation in R, see Van der Ark (2012a). For ask==FALSE, the default graphic device in R may only display the last graph.

Value

Returns a graph.

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Ligtvoet, R., L. A. van der Ark, J. M. te Marvelde, and K. Sijtsma (2010). Investigating an invariant item ordering for polytomously scored items. *Educational and Psychological Measurement, 70*, 578-595.

Ligtvoet, R., L. A. van der Ark, W. P. Bergsma, and K. Sijtsma (2011). Polytomous latent scales for the investigation of the ordering of items. *Psychometrika, 76*, 200-216.

Sijtsma, K., R. R. Meijer, and L. A. van der Ark (2011). Mokken scale analysis as time goes by: An update for scaling practitioners. *Personality and Individual Differences, 50*, 31-37.

Van der Ark, L. A. (2012). New developments in Mokken scale analysis in **R**. *Journal of Statistical Software, 48* (5), 1-27. http://www.jstatsoft.org/v20/i11

Van der Ark, L. A. (2014). Visualizing uncertainty of estimated response functions in nonparametric item response theory. In R. E. Millsap, L. A. van der Ark, D. Bolt, and C. M. Woods (Eds.), *New developments in quantitative psychology* (pp. 59-68). New York: Springer.

See Also

check.iio, summary.iio.class

Examples

```
data(acl)
Communality <- acl[,1:10]
iio.list <- check.iio(Communality)
summary(iio.list)
plot(iio.list)
```

---

plot.monotonicity.class

Plot monotonicity.class objects

Description

S3 Method to plot objects of class monotonicity.class. Graphic display of the checks of monotonicity. One graph for each item plotting the estimated item step response functions and/or item response function, plus confidence envelopes (Van der Ark, 2012).

Usage

```
## S3 method for class 'monotonicity.class'
plot(x, items = all.items, curves = "both", ci = TRUE,
              alpha = .05, color = "black", transparancy = 20, ask = TRUE, ...)
```

Arguments

x	Object of class monotonicity.class produced by check.monotonicity.
items	vector containing the numbers of the items for which the results are depicted graphically. Default the results for all items are depicted.
curves	"ISRF": Item step response functions (ISRFs) are depicted; "IRF": item response functions (IRFs) are depicted; "both": Both ISRFs and IRFs are depicted. Default is "both"
ci	Boolean. If TRUE (default), then confidence envelops are plotted around IRFs and ISRFs.
alpha	Type of plotted (1 - alpha) confidence intervals. By default 95-percent confidence intervals are depicted
color	Color of the plotted curves and confidence envelops. Default is black.

`transparancy`	Transparancy of the confidence intervals. Higher values result in more opaque colors for the confidence intervals.
`ask`	Boolean. If `TRUE` (default), then `par("ask"=TRUE)`; i.e., a hard return between subsequent plots is required. If `FALSE`, then `par("ask"=FALSE)`.
`...`	Optional graphical parameters will be ignored

Details

For details of the method, see Molenaar and Sijtsma (2000) and Sijtsma and Molenaar (2002). For details of the confidence envelopes, see Van der Ark (2012) For the implementation in R, see Van der Ark (2007). For `curves=="both"`, both plots are plotted simultaneously using `layout(matrix(c(1,2)1,2))`. For `ask=="FALSE"`, the default graphic device in R may only display the last graph.

Value

Returns a graph.

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

Van der Ark, L. A. (2014). Visualizing uncertainty of estimated response functions in nonparametric item response theory. In R. E. Millsap, L. A. van der Ark, D. Bolt, and C. M. Woods (Eds.), *New developments in quantitative psychology* (pp. 59-68). New York: Springer.

See Also

`check.monotonicity`, `summary.monotonicity.class`

Examples

```
data(acl)
Communality <- acl[,1:10]
monotonicity.list <- check.monotonicity(Communality)
plot(monotonicity.list)
summary(monotonicity.list)
```

`plot.pmatrix.class` *Plot pmatrix.class objects*

Description

S3 Method to `plot` objects of class pmatrix.class. Graphic display of the checks of pmatrix. One graph for each item plotting the rows of the P(++) matrix and rows of the P(–) matrix. If nonintersection holds the lines in the plots of the P(++) matrix must be nondecreasing and the lines in the plots of the P(–) matrix nust be nonincreasing.

Usage

```
## S3 method for class 'pmatrix.class'
plot(x, items = all.items, pmatrix = "both", ci = TRUE,
              alpha = .05, color = "black", transparancy = 20, ask = TRUE, ...)
```

Arguments

`x`	Object of class pmatrix.class produced by `check.pmatrix`.
`items`	vector containing the numbers of the item pairs for which the results are depicted graphically. Default the results for all items are depicted.
`pmatrix`	Valid options are `"ppp"`, `"pmm"`, and `"both"`. If `pmatrix="ppp"`, then the P(++) matrix is plotted, if `pmatrix="pmm"`, then the P(–) matrix is plotted, if `pmatrix="both"`, then both the P(++) matrix and P(–) matrix are plotted.
`ci`	Boolean. If `TRUE`, then confidence envelops are plotted around IRFs and ISRFs.
`alpha`	Type of plotted (1 - alpha) confidence intervals. By default 95-percent confidence intervals are depicted
`color`	Color of the plotted curves and confidence envelops. Default is black.
`transparancy`	Transparancy of the confidence intervals. Higher values result in more opaque colors for the confidence intervals.
`ask`	Boolean. If `TRUE` (default), then `par("ask"=TRUE)`; i.e., a hard return between subsequent plots is required. If `FALSE`, then `par("ask"=FALSE)`.
`...`	Optional graphical parameters will be ignored

Details

The default graphic device in R may only display the last graph.

In the plot of the P(++) matrix and the P(–) matrix, the x-axis contains the $k = (J-1)m$ item steps not pertaining to item j in order of popularity (ascending). Let $Y_g = 1$ indicate that the g-th item step has been taken: i.e. $Xi \geq y$ and let $Y_g = 0$ indicate that the g-th item step has not been taken: i.e. $Xi < y$. The m lines in the plot of the P(++) matrix connect $P(X_j \geq x, Y_1 = 1), \ldots, P(X_j \geq x, Y_k = 1)), x = 1, \ldots m$. The m lines in the plot of the P(–) matrix connect $P(X_j < x, Y_1 = 0), \ldots, P(X_j < x, Y_k = 0)), x = 1, \ldots, m$.

If the number of item steps on the x-axis is greater than 10, then the labels are not displayed,

Value

Returns a graph.

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

See Also

check.pmatrix, summary.pmatrix.class

Examples

```
data(acl)
Communality <- acl[,1:10]
pmatrix.list <- check.pmatrix(Communality)
plot(pmatrix.list)
summary(pmatrix.list)
```

---

plot.restscore.class *Plot restscore.class objects*

Description

S3 Method to plot objects of class restscore.class. Graphic display of the checks of restscore. One graph for each item pair plotting the estimated item step response functions (ISRFs); confidence envelopes are optional. Intersections of the lines indicate violations of nonintersection,

Usage

```
## S3 method for class 'restscore.class'
plot(x, item.pairs = all.pairs, ci = TRUE, alpha = .05,
              color = c("black", "blue"), transparancy = 20, ask = TRUE, ...)
```

Arguments

x	Object of class restscore.class produced by check.restscore.
item.pairs	vector containing the numbers of the item pairs for which the results are depicted graphically. For example, item.pairs = 1 prints the results for items 1 and 2, item.pairs = 2 prints the results for items 1 and 3, item.pairs = J prints the results for items 1 and J, and item.pairs = J+1 prints the results for items 2 and 3. Default the results for all item pairs are depicted.
ci	Boolean. If TRUE (default), then confidence envelops are plotted around ISRFs.
alpha	Type of plotted (1 - alpha) confidence intervals. By default 95-percent confidence intervals are depicted
color	Color of the plotted lines and confidence envelops. Defaults are black for the first item and blue for the second item.
transparancy	Transparancy of the confidence intervals. Higher values result in more opaque colors for the confidence intervals.
ask	Boolean. If TRUE (default), then par("ask"=TRUE); i.e., a hard return between subsequent plots is required. If FALSE, then par("ask"=FALSE).
...	Optional graphical parameters will be ignored

Details

For details of the method, see Molenaar and Sijtsma (2000) and Sijtsma and Molenaar (2002). For details of the confidence envelopes, see Van der Ark (2012) For the implementation in R, see Van der Ark (2007). For ask==FALSE, the default graphic device in R may only display the last graph. The default number of plots can increase rapidly for large numbers of items.

Value

Returns a graph.

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

Van der Ark, L. A. (2014). Visualizing uncertainty of estimated response functions in nonparametric item response theory. In R. E. Millsap, L. A. van der Ark, D. Bolt, and C. M. Woods (Eds.), *New developments in quantitative psychology* (pp. 59-68). New York: Springer.

See Also

check.restscore, summary.restscore.class

Examples

```
data(acl)
Communality <- acl[,1:10]
restscore.list <- check.restscore(Communality)
plot(restscore.list)
summary(restscore.list)
```

---

recode *Recodes negatively worded items*

Description

Returns a matrix or data.frame with the indicated items recoded.

Usage

```
recode(X, items = NULL, values = defaultValues)
```

Arguments

X	matrix or data frame of numeric data containing the responses of nrow(X) respondents to ncol(X) items. Missing values are allowed
items	Vector of integers indicating the items to be recoded
values	Vector of possible item scores. By default the range of the observed values is taken

Details

The result is X for which columns items have been recoded.

Value

The result is X for which columns items have been recoded.

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

See Also

DS14, twoway

Examples

```
data(DS14)

# Handle missing data and recode negatively worded items
X <- DS14[, 3 : 16]
X <- twoway(X)
X <- recode(X, c(1, 3))
head(X)
```

`summary.iio.class` *Summarize iio.class objects*

Description

S3 Method for `summary` of objects of class iio.class. Summarize checks of invariant item ordering.

Usage

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

Arguments

`object`	list produced by `check.iio`
`...`	Optional parameters will be ignored

Value

`method`	String describing the method used for investigating invariant item ordering: Either "MIIO" (Method Manifest Invariant Item Ordering), "MSCPM" (Method Manifest Scale Cumulative Probability Model), or "IT" (Method Increasingness in Transposition)
`item.summary`	Matrix with `ncol(X)` rows and 10 columns, showing for each item a summary of the violations of an invariant item ordering: `itemH` = Item-scalability coefficient; `#ac` = number of active pairs that were investigated; `#vi` = number of violations in which the item is involved; `#vi/#ac` = propotion of active pairs that is involved in a violation; `maxvi` = maximum violation; `sum` = sum of all violations; `tmax` (for method MIIO), `zmax` (for method MSCPM), or `xmax` (for method IT) = maximum t-value, z-value, and chi-square value, respectively; `tsig` (for method MIIO), `zsig` (for method MSCPM), or `xsig` (for method IT) = number of significant t-values, z-values, and chi-square values, respectively; `crit` = Crit value (Molenaar and Sijtsma, 2000, pp. 49, 74).
`backward.selection`	Matrix showing the number of violations for each item (rows) at each step of the backward item selection proces (columns). The first column shows the number of violations for each item. Then in an iterative procedure, the item whose removal results in the largest decrease of violations is removed and the number of

violations is computated again. If the reduction in the number of violations is undecisive then, from the candidate items, the item having the smallest scalability coefficient is removed. The backward selection procedure stops when there are no more violations.

HT Numeric: Coefficient HT for the selected items. Given an IIO, coefficient HT expresses the strength of the ordering (Ligtvoet et al., 2010).

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Ligtvoet, R., L. A. van der Ark, J. M. te Marvelde, and K. Sijtsma (2010). Investigating an invariant item ordering for polytomously scored items. *Educational and Psychological Measurement, 70*, 578-595.

Ligtvoet, R., L. A. van der Ark, W. P. Bergsma, and K. Sijtsma (2011). Polytomous latent scales for the investigation of the ordering of items. *Psychometrika, 76*, 200-216.

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K., R. R. Meijer, and L. A. van der Ark (2011). Mokken scale analysis as time goes by: An update for scaling practitioners. *Personality and Individual Differences, 50*, 31-37.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software, 20* (11), 1-19. http://www.jstatsoft.org/v20/i11

Van der Ark, L. A. (2012). New developments in Mokken scale analysis in R. *Journal of Statistical Software, 48*(5), 1-27. http://www.jstatsoft.org/v48/i5

See Also

check.iio,plot.iio.class

Examples

```
# Examples from Ligtvoet et al. (2011).

data(cavalini)
X1 <- cavalini[,c(3,5,6,7,9,11,13,14)]

# Use Method MIIO and remove items violating MIIO
iio.list1 <- check.iio(X1)
summary(iio.list1)
plot(iio.list1)
X2 <- X1[,is.na(charmatch(dimnames(X1)[[2]],names(iio.list1$items.removed)))]

# Use Method MSCPM and remove items violating MSCPM
iio.list2 <- check.iio(X2,method="MSCPM")
summary(iio.list2)
X3 <- X2[,is.na(charmatch(dimnames(X2)[[2]],names(iio.list2$items.removed)))]
```

```
# Use Method IT
iio.list3 <- check.iio(X3,method="IT")
summary(iio.list3)
```

`summary.monotonicity.class`
Summarize monotonicity.class objects

Description

S3 Method for `summary` of objects of class monotonicity.class. Summarizes checks of monotonicity

Usage

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

Arguments

`object`	list produced by `check.monotonicity`
`...`	Optional parameters will be ignored

Value

Matrix with `ncol(X)` rows and 10 columns, showing for each item a summary of the violations of monotonicity: `itemH` = Item-scalability coefficient; `#ac` = number of active pairs that were investigated; `#vi` = number of violations in which the item is involved; `#vi/#ac` = propotion of active pairs that is involved in a violation; `maxvi` = maximum violation; `sum` = sum of all violations; `zmax` = maximum z-value; `zsig` = number of significant z-values; `crit` = Crit value (Molenaar and Sijtsma, 2000, pp. 49, 74).

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Mokken, R. J. (1971) *A Theory and Procedure of Scale Analysis*. Berlin, Germany: De Gruyter.

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

See Also

check.monotonicity, plot.monotonicity.class

Examples

```
data(acl)
Communality <- acl[,1:10]
monotonicity.list <- check.monotonicity(Communality)
plot(monotonicity.list)
summary(monotonicity.list)
```

---

summary.pmatrix.class *Summarize pmatrix.class objects*

Description

S3 Method for summary of objects of class pmatrix.class. Summarize checks of nonintersection using method pmatrix.

Usage

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

Arguments

object	list produced by check.pmatrix
...	Optional parameters will be ignored

Details

For obtaining the P++ and P– matrix, see examples.

Value

Matrix with ncol(X) rows and 10 columns, showing for each item a summary of the violations of nonintersection using method pmatrix: itemH = Item-scalability coefficient; #ac = number of active pairs that were investigated; #vi = number of violations in which the item is involved; #vi/#ac = propotion of active pairs that is involved in a violation; maxvi = maximum violation; sum = sum of all violations; zmax = maximum z-value; zsig = number of significant z-values; crit = Crit value (Molenaar and Sijtsma, 2000, pp. 49, 74).

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Mokken, R. J. (1971) *A Theory and Procedure of Scale Analysis*. Berlin, Germany: De Gruyter.

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

See Also

check.pmatrix, plot.pmatrix.class

Examples

```
data(acl)
Communality <- acl[,1:10]
pmatrix.list <- check.pmatrix(Communality)
plot(pmatrix.list)
summary(pmatrix.list)

# Small example showing how to retrieve the P++ matrix and the P-- matrix
SmallExample <- acl[,1:4]
pmatrix.list <- check.pmatrix(SmallExample)
pmatrix.list$results$Ppp
pmatrix.list$results$Pmm
```

---

summary.restscore.class

Summarize restscore.class objects

Description

S3 Method for summary of objects of class restscore.class. Summarize checks of nonintersection using method restscore.

Usage

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

Arguments

object	list produced by check.restscore
...	Optional parameters will be ignored

Value

Matrix with ncol(X) rows and 10 columns, showing for each item a summary of the violations of nonintersection using method restscore: itemH = Item-scalability coefficient; #ac = number of active pairs that were investigated; #vi = number of violations in which the item is involved; #vi/#ac = propotion of active pairs that is involved in a violation; maxvi = maximum violation; sum = sum of all violations; zmax = maximum z-value; zsig = number of significant z-values; crit = Crit value (Molenaar and Sijtsma, 2000, pp. 49, 74).

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Mokken, R. J. (1971) *A Theory and Procedure of Scale Analysis*. Berlin, Germany: De Gruyter.

Molenaar, I.W. and Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. Groningen, The Netherlands: IEC ProGAMMA.

Sijtsma, K, and Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Thousand Oaks, CA: Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. http://www.jstatsoft.org/v20/i11

See Also

check.restscore, plot.restscore.class

Examples

```
data(acl)
Communality <- acl[,1:10]
restscore.list <- check.restscore(Communality)
plot(restscore.list)
summary(restscore.list)
```

---

transreas *Transitive Reasoning*

Description

Data came from 12 dichtomous items administered to 425 children in grades 2 through 6 (Verweij, Sijtsma, and Koops, 1996). Each item is a transitive reasoning task.

Usage

```
data(transreas)
```

Format

A 425 by 13 (grade and scores on 12 items) matrix containing integers. `attributes(transreas)` gives details on the items

Details

Items have two ordered answer categories, *incorrect* (score 0), *correct* (1). (for detailed information, see Sijtsma and Molenaar, 2002, p. 33).

Item	Task	Property	Format	Objects	Measures
T09L	9	length	YA = YB < YC = YD	sticks	12.5, 12.5, 13, 13 (cm)
T12P	12	pseudo			
T10W	10	weight	YA = YB < YC = YD	balls	60, 60, 100, 100 (g)
T11P	11	pseudo			
T04W	4	weight	YA = YB = YC = YD	cubes	65 (g)
T05W	5	weight	YA < YB < YC	balls	40, 50, 70 (cm)
T02L	2	length	YA = YB = YC = YD	tubes	12 (cm)
T07L	7	length	YA > YB = YC	sticks	28.5, 27.5, 27.5 (cm)
T03W	3	weight	YA > YB > YC	tubes	45, 25, 18 (g)
T01L	1	length	YA > YB > YC	sticks	12, 11.5, 11 (cm)
T08W	8	weight	YA > YB = YC	balls	65, 40, 40 (g)
T06A	6	area	YA > YB > YC	discs	7.5, 7, 6.5 (diameter; cm)

References

Verweij, A. C., Sijtsma, K., and Koops, W. (1996). A Mokken scale for transitive reasoning suited for longitudinal research. *International Journal of Behavioral Development, 23*, 241-264.

Sijtsma, K. and Molenaar, I. W. (2002). *Introduction to nonparametric item response theory*, Thousand Oaks, CA: Sage.

Examples

```
# Construction of Table 3.1 in Sijtsma and Molenaar (2002, p. 33)
data(transreas)
grades <- transreas[,1]
item.scores <- transreas[,-1]
Total.group <- round(apply(item.scores,2,mean),2)
for (i in 2:6) assign(paste("Grade.",i,sep=""),
   round(apply(item.scores[grades==i,],2,mean),2))
Task <- c(9,12,10,11,4,5,2,7,3,1,8,6)
Property <- attributes(transreas)$property
Format <- attributes(transreas)$format
Objects <- attributes(transreas)$objects
Measures <- attributes(transreas)$measures
Table.3.1 <- data.frame(Task,Property,Format,Objects,Measures,
   Total.group,Grade.2,Grade.3,Grade.4,Grade.5,Grade.6)
Table.3.1
```

twoway *Two-way imputation*

Description

Returns a single or multiple completed data sets using two-way imputation with normally distributed errors.

Usage

```
twoway(X, nCompletedDataSets = 1, minX = defaultMinX, maxX = defaultMaxX, seed = FALSE)
```

Arguments

X	matrix or data frame of integer data containing the score of now(X) respondents to nicol(X) items. Typically X contains missing values.
nCompletedDataSets	Number of completed data sets.
minX	Minimum item score. By default, the minimum item score is the lowest score found in the data.
maxX	Maximum item score. By default, the maximum item score is the highest score found in the data.
seed	Seed for random sampling. If seed = FALSE (default), no seed is given, otherwise seed must be a numeric value. Replications having the same seed result in exactly the same outcome value.

Details

For single imputation (nCompletedDataSets == 1, default) the function returns an object of the same class as X, for multiple imputation (nCompletedDataSets > 1) the function returns a list. References for two-way imputation include Bernaards and Sijtsma (2000), Sijtsma and Van der Ark (2003), and Van Ginkel, Van der Ark, and Sijtsma (2007).

Value

The result is X for which the missing values have been replaced by imputed values. For multiple imputations, the result is a list of matrices/data frames. For single imputations, the result is a matrix/data frame.

Author(s)

L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Bernaards, C. A. and Sijtsma, K. (2000). Influence of simple imputation and EM methods on factor analysis when item nonresponse in questionnaire data is nonignorable *Multivariate Behavioral Research, 35*, 321-364.

Sijtsma, K. and Van der Ark, L. A. (2003). Investigation and treatment of missing item scores in test and questionnaire data. *Multivariate Behavioral Research, 38*, 505-528.

Van Ginkel, J. R., Van dec Ark, L. A., and Sijtsma, K. (2007). Multiple imputation of item scores in test and questionnaire data, and influence on psychometric results. *Multivariate Behavioral Research, 42*, 387-414.

See Also

DS14, recode

Examples

```
data(DS14)

# Handle missing data and recode negatively worded items
X <- DS14[, 3 : 16]
X <- twoway(X)
X <- recode(X, c(1, 3))
head(X)
```

Index

∗Topic **aisp**
aisp, 7
∗Topic **bounds**
check.bounds, 10
∗Topic **check.ca**
check.ca, 11
∗Topic **check.norms**
check.norms, 18
∗Topic **datasets**
acl, 5
cavalini, 8
DS14, 27
transreas, 41
∗Topic **iio**
check.iio, 14
plot.iio.class, 28
summary.iio.class, 36
∗Topic **monotonicity**
check.monotonicity, 16
plot.monotonicity.class, 30
summary.monotonicity.class, 38
∗Topic **package**
mokken-package, 2
∗Topic **plot**
plot.iio.class, 28
plot.monotonicity.class, 30
plot.pmatrix.class, 32
plot.restscore.class, 33
summary.restscore.class, 40
∗Topic **pmatrix**
check.pmatrix, 19
plot.pmatrix.class, 32
summary.pmatrix.class, 39
∗Topic **recode**
recode, 35
∗Topic **reliability**
check.reliability, 21
∗Topic **restscore**
check.errors, 13
check.restscore, 22
plot.restscore.class, 33
summary.restscore.class, 40
∗Topic **scalability coefficients**
aisp, 7
coefH, 24
coefZ, 26
∗Topic **summary**
summary.iio.class, 36
summary.monotonicity.class, 38
summary.pmatrix.class, 39
∗Topic **twoway**
twoway, 43

acl, *2*, 5
aisp, *3, 4*, 7, *26*

cavalini, *2*, 8
check.bounds, 10
check.ca, *4*, 11, *14*
check.errors, *3, 4*, 13, *16*, *18*, *20*, *22, 23*
check.iio, *3, 4*, *8, 9*, *14*, 14, *18*, *20*, *22, 23*, *29, 30*, *36, 37*
check.monotonicity, *3*, *8*, *14*, *16*, 16, *20*, *22, 23*, *30, 31*, *38, 39*
check.norms, *4*, 18
check.pmatrix, *3*, *8*, *14*, *16*, *18*, 19, *22, 23*, *32, 33*, *39, 40*
check.reliability, *3*, *8*, *14*, *16*, *18*, *20*, 21, *23*
check.restscore, *3*, *8*, *16*, *18*, *20*, *22*, 22, *34, 35*, *40, 41*
coefH, *3*, *8*, *14*, *16*, *18*, *20*, *22, 23*, 24, *26*
coefZ, *3*, *25*, 26

DS14, *2*, *4*, *12*, *19*, 27, *35*, *44*

mokken, *3*
mokken *(*mokken-package*)*, 2
mokken-package, 2

plot.iio.class, *4*, *16*, 28, *37*
plot.monotonicity.class, *4*, *18*, 30, *39*
plot.pmatrix.class, *20*, 32, *40*
plot.restscore.class, *4*, *14*, *23*, 33, *41*

recode, *4*, *12*, *19*, *28*, 35, *44*

search *(*aisp*)*, 7
search.normal, *3*, *25*
sort, *24*
summary.iio.class, *16*, *30*, 36
summary.monotonicity.class, *18*, *31*, 38
summary.pmatrix.class, *20*, *33*, 39
summary.restscore.class, *14*, *23*, *35*, 40
Sys.getlocale, *24*

transreas, *2*, 41
twoway, *4*, *12*, *19*, *28*, *35*, 43