Package ‘mokken’

June 18, 2024

Version 3.1.2

Date 2024-6-17

Title Conducts Mokken Scale Analysis

Author L. Andries van der Ark [aut, cre],
Letty Koopman [aut],
J. Hendrik Straat [ctb],
Don van den Bergh [ctb]

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

Depends R (>= 3.5.0), 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

LinkingTo Rcpp

Imports Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-06-18 08:50:02 UTC

Contents

`mokken-package` *Mokken Scale Analysis*

Description

Mokken scale analysis (Mokken, 1971; Sijtsma & Molenaar, 2002; Sijtsma & Van der Ark, 2017) 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 & Sijtsma, 2000).

Details

Package:	mokken
Type:	Package
Version:	3.1.2
Date:	2024-06-17
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.
`autonomySupport`	Scores from students on their teacher's autonomy support
`balance`	Scores on balance taskts
`cavalini`	Scores on an inventory on industrial malodor
`DS14`	Scores on a Type D test (bootstrap sample)
`mcmi`	Scores on some items from the Dutch version of the Millon Clinical Multiaxial Inventory
`SWMD`	Scores from pupils nested in classrooms on their well-being with teachers
`SWMDK`	Scores from pupils nested in classrooms on their well-being with teachers and classmates
`transreas`	Scores on a transitive reasoning test
`transreas2`	More scores on a transitive reasoning test
`trog`	Scores from children on the clustered items of the Norwegian adaptation of the Test for Reception of Gra

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. The Mokken package was created by Andries van der Ark, who is also the maintainer. Significant parts have been developed by Letty Koopman (multilevel and clustered item analysis) and Don van den Berg and Hendrik Straat (all C and C++ codes). Thanks are due to Michael Allerhand, Geert H. van Kollenburg, Renske E. Kuijpers, Rudy Ligtvoet, Hannah E. M. Oosterhuis, Daniel W. van der Palm, and Max Welz for contributing R code; to Geert H. van Kollenburg, Patrick Mair, and Don van Ravenswaaij for testing the software; to Wijbrandt van Schuur for comments on the vignette; to Michael Allerhand, 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, Stefan Vermeent, Roger Watson, Stefanie Wind, Max Welz, and Na Yang for reporting comments or bugs; to Diederick Stoel (ProfitWise) for financial support, to Samantha Bouwmeester, Pierre Cavalini, Johan Denollet, Gina Rossi, Harrie C. M. Vorst, Ellen Iren Brinchmann, 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

The following modifications have been made in Version 2.8.4 in comparison to previous versions.

check.iio	New code for computing HT for large samples

The following modifications have been made in Version 2.8.5 in comparison to previous versions.

MLcoefH	New code for computing two-level scalability coefficients and standard errors
autonomySupport	New two-level data set.

The following modifications have been made in Version 2.8.9 and 2.8.10 in comparison to previous versions.

coefH	Included possibility to include a fixed item-step order
MLcoefH	Code updated
check.errors	Code updated

The following modifications have been made in Version 2.8.11 in comparison to previous versions.

plot	The level of tranparancy of the plotted confidence intervals can be adjusted manually
MLcoefH	Code updated

The following modifications have been made in Version 2.8.12 in comparison to previous versions.

check.monotonicity	Z statistic adjusted (Molenaar & Sijtsma, 2000. p. 72)
check.norms	Z Output corrected for `nice.output = FALSE`

The following modifications have been made in Version 2.9.0 in comparison to previous versions.

coefH	Z Solution of Koopman et al. (2017) implemented to solve the problem of equal item steps and code updated
MLcoefH	Z Solution of Koopman et al. (2017) implemented to solve the problem of equal item steps and code updated

The following modifications have been made in Version 3.0.0 in comparison to previous versions.

aisp	Genetic algorithm has been reprogrammed and is now much faster.
	Argument lowerbound can now be a vector, enabling the investigation of several lower bounds simultaneously.
	Extra argument to specify which standard errors should be used in Z-test (Koopman et al., 2020).
	Extra argument to indicate which null-hypothesis should be used to test Hi (i.e., Hi = c or Hi = 0)(Koopman et a
	Extra argument to handle nested data (Koopman et al., 2020).
check.iio	Computation of Coefficient HT for large samples is now much faster.
coefH	New standard errors for nested data (Koopman et al. in press a).
coefZ	Extra argument to compute the Z-score using lowerbound as the null hypothesis (Koopman et al., 2020).
	Extra argument to compute Z-score using delta method standard error, but the original method remains available
	Extra argument to compute Z-score in nested data (Koopman et al., 2020).
ICC	New function for ICCs in two-level Mokken scale analysis (Koopman et al. in press a)
MLcoefH	Extra argument for weighted proportions. Reduces bias in two-level standard errors (Koopman et al. in press a)
	Extra argument for a fixed item-step order
SWMD	New data file (Koopman et al. in press a)

The following modifications have been made in Version 3.0.3 in comparison to previous versions.

coefZ	Error handling added
	The delta test uses range-preserving asymptotic theory (Koopman,et al., in press b)
coefH	Range-preserving confidence intervals added (Koopman et al., in press b)
	Extra argument to print variance-covariance matrices of estimated coefficients
	Error handling added for more than 10 response categories
MLcoefH	Range-preserving confidence intervals added (Koopman et al., in press b)
	Extra argument to print variance-covariance matrices of estimated coefficients
MLcoefZ	New function for z-scores of two-level scalability coefficients (Koopman et al., in press b)
aisp	type.se default adjusted and additional error handling added
check.errors	Repaired bug in check.errors
mcmi	New data file (Sijtsma & van der Ark, 2020)
All functions	A warning has been added if items have different numbers of response categories

The following modifications have been made in Version 3.0.4 in comparison to previous versions.

aisp	Argument type.z replaces type.se to accommodate three types of z scores (Mokken's Z, Wald-based, and ra
search.normal	Argument type.z replaces type.se to accommodate three types of z scores (Mokken's Z, Wald-based, and ra
coefZ	Argument type.z replaces type.se to accommodate three types of z scores (Mokken's Z, Wald-based, and ra
MLcoefZ	Argument type.z is added to accommodate two types of z scores (Wald-based and range-preserving)
coefH	Argument type.ci is added to accommodate two types of confidence intervals (Wald-based and range-preser
	Argument print.to.screen is replaced by results
MLcoefH	Argument type.ci is added to accommodate two types of confidence intervals (Wald-based and range-preser
SWMDK	New data file (Koopman et al., in press a)
All functions	A warning has been added if items have different numbers of response categories

The following modifications have been made in Version 3.1.0 in comparison to previous versions.

check.monotonicity	Condition N > 500 to determined the default value of minsize has been changed to N >= 50
check.restscore	Condition N > 500 to determined the default value of minsize has been changed to N >= 50

check.iio	Condition N > 500 to determined the default value of minsize has been changed to N >= 50
check.iio	For dichtomous items, a z-test is used (rather than a t-test) to test violations of manifest inv
check.iio	For polytomous items, a paired t-test (rather than independent two-sample t-test) is now pe
check.monotonicity	Argument level.two.var is added to enable two-level model fit checks (Koopman et al., 202
summary.monotonicity.class	Adjusted to handle two-level fit results from check.monotonicity.
plot.monotonicity.class	Adjusted to handle two-level fit results from check.monotonicity.
check.iio	Argument level.two.var is added to enable two-level model fit checks (Koopman et al., 202
summary.iio.class	Adjusted to handle two-level fit results from check.iio.
plot.iio.class	Adjusted to handle two-level fit results from check.iio.

The following modifications have been made in Version 3.1.1 in comparison to previous versions.

check.iio	A bug (resulting in an error) that occurred if the sample size was exactly a multiple of 1,000 was fixed.
check.iio	Argument fixed.item.order is added to enable confirmatory analysis of a given item order, and to allow for inve
All functions	A tibble class input is now allowed in addition to matrix and data frames.
trog	New data file (Koopman & Braeken, 2024).

The following modifications have been made in Version 3.1.1 in comparison to previous versions.

ICC	The example on the helpfile of the ICC function was updated.

Author(s)

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

References

Koopman, L. & Braeken, J. (2024). Investigating the Ordering Structure of Clustered Items Using Nonparametric Item Response Theory. Manuscript submitted for publication.

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023a). Assumptions and Properties of Two-Level Nonparametric Item Response Theory Models. Manuscript submitted for publication.

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023b). Evaluating Model Fit in Two-Level Mokken Scale Analysis. *Psych* 5(3), 847-865. doi:10.3390/psych5030056

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2020). New advances in Mokken scale analysis. Paper presented at the online meeting of the Psychometric Society, July, 2020.

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press a). A two-step, test-guided Mokken scale analysis for nonclustered and clustered data. *Quality of Life Research*. (advanced online publication) doi:10.1007/s11136021028402

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press b). Range-preserving confidence intervals and significance tests for scalability coefficients in Mokken scale analysis. In M. Wiberg, D. Molenaar, J. Gonzalez, & Kim, J.-S. (Eds.), *Quantitative Psychology; The 1st Online Meeting of the Psychometric Society, 2020*. Springer. doi:10.1007/9783030747725_16

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

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

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Sijtsma, K., & Van der Ark, L. A. (2017). A tutorial on how to do a Mokken scale analysis on your test and questionnaire data. *British Journal of Mathematical and Statistical Psychology, 70*, 137-158. doi:10.1111/bmsp.12078

Sijtsma, K., & Van der Ark, L. A. (2020), *Measurement models for psychological attributes.* Boca Raton, FL: Chapman and Hall/CRC Press. `https://www.routledge.com/Measurement-Models-for-Psychological-At Sijtsma-Ark/p/book/9780367424527`

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software, 20*(11), 1-19. doi:10.18637/jss.v020.i11

Van der Ark, L. A. (2010). Getting started with Mokken scale analysis in **R**. Unpuablished 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. doi:10.18637/jss.v048.i05

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., & Heilbrun,A. B. (1980) *The Adjective Check List, Manual 1980 Edition*. Consulting Psychologists Press.

Van der Ark, L. A. (2007) Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v020.i11

Examples

```
data(acl)
```

---

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

Description

Returns a matrix with as many rows as there are items, indicating to which scale an item belongs for each lowerbound.

Usage

```
aisp(X, lowerbound=.3, search="normal", alpha=.05, StartSet=FALSE, popsize=20,
     maxgens=default.maxgens, pxover=0.5, pmutation=0.1, verbose=FALSE,
     type.z = "Z", test.Hi = FALSE, level.two.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
`search`	Type of item selection procedure: "normal": Mokken's automated item selection procedure (Mokken, 1971; Molenaar & Sijtsma, 2000; Sijtsma & Molenaar, 2002); "ga": item selection using a genetic algorithm (Straat, van der Ark, & Sijtsma, 2013). The default is "normal".
`lowerbound`	Value or vector with numeric scaling criteria; 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`.
`type.z`	Indicates which type of Z-test is used to evaluate whether coefficients meet the scaling criteria: "WB": Wald-based z-score based on standard errors as approximated by the delta method (Kuijpers et al., 2013; Koopman et al., in press a); "RP": Range-preserving z-score, also based on the delta method (Koopman, et al., in press b); "Z": uses original Z-test (Mokken, 1971; Molenaar & Sijtsma, 2000; Sijtsma & Molenaar, 2002). The default is "Z", but is changed to "WB" for test.Hi == TRUE or if a level.two.var is given.
`test.Hi`	If `FALSE`: tests if Hi is significantly larger than zero; If `TRUE` tests if Hi is significantly larger than `lowerbound`. The default is `FALSE`.
`level.two.var`	vector of length `nrow(X)` or matrix with number of rows equal to `nrow(X)` that indicates the level two variable for nested data (Koopman et al., in press a).

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 matrix with J rows. 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, L. Koopman

References

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press a). A two-step, test-guided Mokken scale analysis for nonclustered and clustered data. *Quality of Life Research*. (advanced online publication) doi:10.1007/s11136021028402

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press b). Range-preserving confidence intervals and significance tests for scalability coefficients in Mokken scale analysis. In M. Wiberg, D. Molenaar, J. Gonzalez, & Kim, J.-S. (Eds.), *Quantitative Psychology; The 1st Online Meeting of the Psychometric Society, 2020*. Springer. doi:10.1007/9783030747725_16

Kuijpers, R. E., Van der Ark, L. A., & Croon, M. A. (2013). Standard errors and confidence intervals for scalability coefficients in Mokken scale analysis using marginal models. *Sociological Methodology, 43,* 42-69. doi:10.1177/0081175013481958

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

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

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Straat, J. H., Van der Ark, L. A., & Sijtsma, K. (2013). Comparing optimization algorithms for item selection in Mokken scale analysis. *Journal of Classification, 30,* 72-99. doi:10.1007/s00357013-9122y

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v020.i11

Van der Ark, L. A. (2012). New developments in Mokken scale analysis in **R**. *Journal of Statistical Software*, 48. doi:10.18637/jss.v048.i05

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], se = FALSE)

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

# Perform aisp for increasing lowerbounds

 scales <- aisp(Communality, lowerbound = seq(0, .55, .05))
 scales


# Use a significant test for criteria Hi > c (rather than the point estimate)
scale <- aisp(Communality, type.z = "WB", test.Hi = TRUE, verbose = TRUE)
coefH(Communality[,scale==1])

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

# Perform aisp on two-level data
data(autonomySupport)
scores <- autonomySupport[, -1]
classes <- autonomySupport[, 1]
 scale <- aisp(scores, type.z = "WB", level.two.var = classes)
 coefH(scores[, scale==1], level.two.var = classes)
```

`autonomySupport` *Autonomy Support Data*

Description

A two-level dataset with scores of 14 teachers who are rated by a group of pupils on 7 items from a Dutch Autonomy Support questionnaire (group size ranged between 5 and 39 pupils, total number of pupils is 259).

Usage

```
data(autonomySupport)
```

Format

A 259 by 8 data frame containing integers. The first column reflects a teacher indicator, the remaining columns the 7 items, see `colnames(autonomySupport)`.

Details

Each item has five ordered answer categories from *not at all/never* (score 1) to *certainly/always* (score 5). The items reflect several autonomy supportive behaviours from teachers.

Item	*Short*	*Content*
Item 1	Choose	The teacher lets me choose what I am going to do
Item 2	Decide	The teacher decides which task I will start with (inversely coded)
Item 3	Task	I get to choose which task I will start with
Item 4	Listen	The teacher listens to me when I disagree with something
Item 5	Help	The teacher helps me when I ask for it
Item 6	Accept	The teacher accepts me for who I am
Item 7	Understand	The teacher helps me when I do not understand a task

Source

The seven items are a subset from a self-constructed 27-item questionnaire on teacher's autonomy support. Data were collected and made available by L. Koopman from the University of Amsterdam.

References

Koopman, L., Zijlstra, B. J. H. & Van der Ark, L. A., (2019). Standard errors of two-lvel scalability coefficients. *British Journal of Statistical and Mathematical Psychology, 73,* 213-236. doi:10.1111/bmsp.12174

See Also

`MLcoefH`,

Examples

```
data(autonomySupport)
```

`balance` *Balance Data*

Description

Scores of 484 todlers on 25 balance-problem items.

Usage

```
data(balance)
```

Format

A 484 by 25 data frame containing integers.

Details

The items include 5 conflict-balance (CB) items , 5 conflic-distance (CD) items, 5 conflict-weight (CW) items, 5 distance (D) items, and 5 weight (W) items. Respondents have been deidentified, and covariates have been removed. The deidentified data do allow to replicate the analyses in Sijtsma and Van der Ark (2020, chapter 5) using the code available from https://osf.io/e9jrz.

Source

The data were collected by Leo van Maanen (see, Van Maanen, Been & Sijtsma, 1989).

References

Van Maanen, L., Been, P. H., & Sijtsma, K. (1989). Problem solving strategies and the Linear Logistic Test Model. In E. E. Ch. I. Roskam (Ed.), *Mathematical psychology in progress* (pp. 267-287). Springer.

Sijtsma, K., & Van der Ark, L. A. (2020), *Measurement models for psychological attributes.* Chapman and Hall/CRC Press. https://www.routledge.com/Measurement-Models-for-Psychological-Attributes/Sijtsma-Ark/p/book/9780367424527

Examples

```
data(balance)
```

`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(cavalini)` 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 & 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., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. 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. doi:10.1007/S1133601393415

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 & 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, & 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., & Sijtsma, K. (2016). Using conditional association to identify locally Independent item sets. *Methodology, 12*, 117-123. doi:10.1027/16142241/a000115

Sijtsma, K., van der Ark, L. A., & 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, & S.-M. Chow (Eds.), *Quantitative psychology research: The 79th Annual Meeting of the Psychometric Society, Madison, Wisconsin, 2014.* (pp. 109 - 120) Springer. doi:10.1007/9783319199771_9

See Also

`DS14`, `recode`, `twoway`

Examples

```
data(DS14)

# Handle missing data nand 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 & 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. doi:10.1177/014662169401800402

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

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

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v020.i11

Zijlstra, W. P., Van der Ark, L. A., & Sijtsma, K. (e2007). Outlier detection in test and questionnaire data. *Multivariate Behavioral Research, 42*, 531-555. doi:10.1080/00273170701384340

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). For a discription of investigating the ordering structure of clustered items, see Koopman & Braeken (2024).

For two-level test data (clustered respondents) argument level.two.var exist, for clustered item data, argument fixed.item.order exist. For both arguments, two lists are returned, containing the results for level 1 (person or item level) and level 2 (group or cluster level), respectively. Only method MIIO is implemented for two-level and clustered-item test data.

Usage

```
check.iio(X, method="MIIO", minvi = default.minvi, minsize = default.minsize,
alpha = .05, item.selection=TRUE, verbose=FALSE, fixed.item.order = NULL,
level.two.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
`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`

`fixed.item.order`
Matrix or vector containing J numeric values to indicate the item ordering from easy to difficult, to perform a confirmatory analysis of manifest invariant item/cluster ordering. For clustered items, the cluster numbers are given, which are repeated for each item of that cluster, such that the length is still J, for example c(1, 1, 2, 2, 3, 3) for three clusters of two items, of which the first cluster is easiest and the last cluster is most difficult (see Koopman & Braeken, 2024).

`level.two.var` Add respondent-clustering variable to get results for Level 1 (person level) and Level 2 (cluster level; see Koopman et al., 2023a,b)

.

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. For an example of clustered items, see Koopman & Braeken (2024).

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. For clustered items, coefficient HBT and ratio HBT/HT is also estimated, see Koopman & Braeken (2024).

`method` The argument `method`

`item.mean` The mean item scores

Author(s)

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

References

Koopman, L. & Braeken, J. (2024). Investigating the Ordering Structure of Clustered Items Using Nonparametric Item Response Theory. Manuscript submitted for publication.

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023a). Assumptions and Properties of Two-Level Nonparametric Item Response Theory Models. Manuscript submitted for publication.

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023b). Evaluating Model Fit in Two-Level Mokken Scale Analysis. Manuscript submitted for publication.

Ligtvoet, R., L. A. van der Ark, J. M. te Marvelde, & K. Sijtsma (2010). Investigating an invariant item ordering for polytomously scored items. *Educational and Psychological Measurement, 70*, 578-595. doi:10.1177/0013164409355697

Ligtvoet, R., L. A. van der Ark, W. P. Bergsma, & K. Sijtsma (2011). Polytomous latent scales for the investigation of the ordering of items. *Psychometrika, 76*, 200-216. doi:10.1007/s11336010-91998

Sijtsma, K., R. R. Meijer, & Van der Ark, L. A. (2011). Mokken scale analysis as time goes by: An update for scaling practitioners. *Personality and Individual Differences, 50*, 31-37. doi:10.1016/j.paid.2010.08.016

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software, 20* (11), 1-19. doi:10.18637/jss.v020.i11

Van der Ark, L. A. (2012). New developments in Mokken scale analysis in R. *Journal of Statistical Software, 48*(5), 1-27. doi:10.18637/jss.v048.i05

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]
summary(check.iio(Autonomy))
plot(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)

# Examples for investigating the ordering structure of a clustered item set
# (Koopman & Braeken, 2024)

data("trog")
clusters <- rep(1:20, each = 4)
ico <- check.iio(trog, item.selection = FALSE, fixed.item.order = clusters)
summary(ico)


# Compute two-level fit statistics (Koopman et al., 2023a, 2023b)

data("autonomySupport")
dat <- autonomySupport[, -1]
groups <- autonomySupport[, 1]
autonomyMIIO <- check.iio(dat, item.selection = FALSE, level.two.var = groups)
summary(autonomyMIIO)
plot(autonomyMIIO)
```

`check.monotonicity` *Check of Monotonicity*

Description

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

For two-level test data (clustered respondents) argument level.two.var exist, such that two lists are returned, containing the results for level 1 (person level) and level 2 (cluster level), respectively. Only method MIIO is implemented for two-level test data.

Usage

```
check.monotonicity(X, minvi = 0.03, minsize = default.minsize, level.two.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
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 $N < 250$
level.two.var	Add respondent-clustering variable to get results for Level 1 (person level) and Level 2 (cluster level; see Koopman et al., 2023a,b)

.

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., & Sijtsma, K. (2000). Latent and manifest monotonicity in item response models. *Applied Psychological Measurement, 24*, 65-81. doi:10.1177/01466216000241004

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023a). Assumptions and Properties of Two-Level Nonparametric Item Response Theory Models. Manuscript submitted for publication.

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023b). Evaluating Model Fit in Two-Level Mokken Scale Analysis. Manuscript submitted for publication.

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

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

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v020.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)

# Compute two-level fit statistics (Koopman et al., 2023a, 2023b)
data("autonomySupport")
dat <- autonomySupport[, -1]
groups <- autonomySupport[, 1]
autonomyMM <- check.monotonicity(dat, level.two.var = groups)
summary(autonomyMM)
plot(autonomyMM)
```

---

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, & Sijtsma, 2017).

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., & Sijtsma, K. (2017). Standard errors and confidence intervals of norm statistics for educational and psychological tests. *Psychometrika, 82*, 559-588. doi:10.1007/s1133601695358

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 & Sijtsma, 2000; Sijtsma & 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 & Sijtsma, 2000; Sijtsma & 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*. De Gruyter.

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

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v048.i05

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 & 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, & 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. doi:10.1007/BF02310555

Guttman, L. (1945). A basis for analyzing test-retest reliability. *Psychometrika, 10*,255-282. doi:10.1007/BF02288892

Molenaar, I. W., & Sijtsma, K. (1984). Internal consistency and reliability in Mokken's nonparametric item response model. *Tijdschrift voor onderwijsresearch, 9*, 257–268. Retrieved from `https://pure.uvt.nl/ws/portalfiles/portal/1030704/INTERNAL.PDF`

Molenaar, I. W., & Sijtsma, K. (1988). Mokken's approach to reliability estimation extended to multicategory items. *Kwantitatieve methoden, 9(28)*, 115-126. Retrieved from `https://pure.uvt.nl/ws/portalfiles/portal/1030575/MOKKEN__.PDF`

Sijtsma, K., & Molenaar, I. W. (1987). Reliability of test scores in nonparametric item response theory. *Psychometrika, 52*,79-97. doi:10.1007/BF02293957

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v020.i11

Van der Ark, L. A. (2010). Computation of the Molenaar Sijtsma statistic. In A. Fink, B. Lausen, W. Seidel, & A. Ultsch (Eds.), *Advances in data analysis, data handling and business intelligence* (pp. 775-784). Springer. doi:10.1007/9783642010446_7

Van der Ark, L. A., Van der Palm, D. W., & Sijtsma, K. (2011). A latent class approach to estimating test-score reliability. *Applied Psychological Measurement, 35*, 380-392. doi:10.1177/0146621610392911

Van der Palm, D. W., Van der Ark, L. A. & Sijtsma, K. (2014). A flexible latent class approach to estimating test-score reliability. *Journal of Educational Measurement, 51*, 339-357. doi:10.1111/jedm.12053

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 & Sijtsma, 2000; Sijtsma & 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*. De Gruyter.

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

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v048.i05

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 & Sijtsma, 2000, pp. 11-13; Sijtsma & Molenaar, chap. 4; Van der Ark, 2007; 2010), as well as their standard errors (Kuijpers et al., 2013; also see Van der Ark et al., 2008) and possibly confidence intervals (Koopman, et al., in press a, in press b). Mokken's coefficients and standard errors can also be estimated in two-level data (Koopman et al., in press a). It is also 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). The estimated variance-covariance matrix of the coefficients is invisible but can be printed by saving the result, see examples.

Usage

```
coefH(X, se = TRUE, ci = FALSE, nice.output = TRUE, level.two.var = NULL,
     group.var = NULL, fixed.itemstep.order = NULL, type.ci = "WB",
     results = TRUE)
```

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
`ci`	The confidence level between 0 and 1 of the range-preserving confidence intervals. If `FALSE` (default), no confidence intervals are printed (Koopman et al.,in press b).
`nice.output`	Logical: If `TRUE`, scalability coefficients and standard errors are combined in an a single object of class `noquote`
`level.two.var`	vector of length `nrow(X)` or matrix with number of rows equal to `nrow(X)` that indicates the level two variable for nested data to get appropriate standard errors (Koopman et al., in press a.
`group.var`	vector of length `nrow(X)` or matrix with number of rows equal to `nrow(X)` to be used as grouping variable

`fixed.itemstep.order`
: matrix with number of rows equal to the number of item steps (m) and number of columns equal to the number of items (J). The matrix should consis the integers `1 : (m * J)`, indicating a predefined order of the items steps with respect to popularity. Value 1 indicates the easiest (most popular) item step, value (m * J) indicates the most difficult item step.

`type.ci`
: If `WB`, Wald-based confidence interval are printed, if `RP` range-preserving confidence intervals are printed (Koopman et al., in press b, in press c). Default is `WB`. Used only if `ci` has been specified.

`results`
: Logical: If TRUE results are printed to the screen. Option `FALSE` is useful only for some internal functions

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 `ci` is specified and `se = TRUE` or `nice.output = FALSE`, there is one additional matrix for the ci's of the Hij coefficients; if `level.two.var` is not null the standard errors are adjusted to take the nesting into account; 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)
`ci.Hij`	confidence intervals of the scalability coefficients of the item pairs (only if `nice.output = FALSE` and/or `se = TRUE`; see details)
`ci.Hi`	confidence intervals of the scalability coefficients of the items (see details)
`ci.H`	confidence intervals of the scalability coefficient of the entire scale (see details)
`Groups`	Scalability coefficients for subgroups (see details)

Author(s)

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

References

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press a). A two-step, test-guided Mokken scale analysis for nonclustered and clustered data. *Quality of Life Research*. (advanced online publication) doi:10.1007/s11136021028402

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press b). Range-preserving confidence intervals and significance tests for scalability coefficients in Mokken scale analysis. In M. Wiberg, D. Molenaar, J. Gonzalez, & Kim, J.-S. (Eds.), *Quantitative Psychology; The 1st Online Meeting of the Psychometric Society, 2020*. Springer. doi:10.1007/9783030747725_16

Kuijpers, R. E., Van der Ark, L. A., & Croon, M. A. (2013). Standard errors and confidence intervals for scalability coefficients in Mokken scale analysis using marginal models. *Sociological Methodology, 43,* 42-69. doi:10.1177/0081175013481958

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*. De Gruyter.

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

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software, 20* (11), 1-19. doi:10.18637/jss.v020.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., & Sijtsma (2008). Mokken scale analysis for dichotomous items using marginal models. *Psychometrika, 73*, 183-208. doi:10.1007/s113360079034z

See Also

coefZ, search.normal

Examples

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

# Compute scalability coefficients and standard errors
Hs <- coefH(Communality)

# Compute scalability coefficients, standard errors, and range-preserving confidence intervals
coefH(Communality, ci = .95)

# Scalability coefficients but no standard errors
coefH(Communality, se = FALSE)

# Scalability coefficients for different groups:
```

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

# Extract variance-covariance matrices
attributes(Hs)
Hs$covHij
Hs$covHi
Hs$covH

# Nested data:
data(autonomySupport)
scores <- autonomySupport[, -1]
classes <- autonomySupport[, 1]
coefH(scores, level.two.var = classes, ci = .95)
```

---

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 using the original method Z (Molenaar and Sijtsma, 2000, pp. 59-62; Sijtsma and Molenaar, p. 40; Van der Ark, 2007; 2010) or the Wald-based method (`WB`) or range-preserving method (`RP`) (Kuijpers et al., 2013; Koopman et al., in press a, in press b). The Wald-based method and range-preserving method can also handle nested data and can test other lowerbounds than zero. Used in the function `aisp`

Usage

```
coefZ(X, lowerbound = 0, type.z = "Z", level.two.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
`lowerbound`	Value of the null hypothesis to which the scalability are compared to compute the Z-score (see details), 0 <= `lowerbound` < 1. The default is 0.
`type.z`	Indicates which type of z-score is computed: "WB": Wald-based z-score based on standard errors as approximated by the delta method (Kuijpers et al., 2013; Koopman et al., in press a); "RP": Range-preserving z-score, also based on the delta method (Koopman et al., in press b); "Z": uses original Z-test and is only appropriate to test lowerbound = 0 (Mokken, 1971; Molenaar and Sijtsma, 2000; Sijtsma and Molenaar, 2002). The default is "Z".
`level.two.var`	vector of length `nrow(X)` or matrix with number of rows equal to `nrow(X)` that indicates the level two variable for nested data (Koopman et al., in press a).

Details

For the estimated item-pair coefficient Hij with standard error $SE(Hij)$, the Z-score is computed as

$$Zij = (Hij - lowerbound)/SE(Hij)$$

if `type.z = "WB"`, and the Z-score is computed as

$$Zij = -(log(1 - Hij) - log(1 - lowerbound))/(SE(Hij)/(1 - Hij))$$

if `type.z = "RP"` (Koopman et al., in press b). For the estimate item-scalability coefficients Hi and total-scalbility coefficients H a similar procedure is used. Standard errors of the Z-scores 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`> L. Koopman

References

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press a). A two-step, test-guided Mokken scale analysis for nonclustered and clustered data. *Quality of Life Research*. (advanced online publication) doi:10.1007/s11136021028402

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press b). Range-preserving confidence intervals and significance tests for scalability coefficients in Mokken scale analysis. In M. Wiberg, D. Molenaar, J. Gonzalez, & Kim, J.-S. (Eds.), *Quantitative Psychology; The 1st Online Meeting of the Psychometric Society, 2020*. Springer. doi:10.1007/9783030747725_16

Kuijpers, R. E., Van der Ark, L. A., & Croon, M. A. (2013). Standard errors and confidence intervals for scalability coefficients in Mokken scale analysis using marginal models. *Sociological Methodology, 43,* 42-69. doi:10.1177/0081175013481958

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

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v020.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]

# Compute the Z-score of each coefficient
coefH(Communality)
coefZ(Communality)

# Using lowerbound .3
coefZ(Communality, lowerbound = .3, type.z = "WB")

# Z-scores for nested data
data(autonomySupport)
scores <- autonomySupport[, -1]
classes <- autonomySupport[, 1]
coefH(scores, level.two.var = classes)
coefZ(scores, type.z = "WB", level.two.var = classes)
```

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 & 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., & 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., & 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. doi:10.1016/j.jad.2015.09.028

Sijtsma, K., & Van der Ark, L. A. (2017). A tutorial on how to do a Mokken scale analysis on your test and questionnaire data. *British Journal of Mathematical and Statistical Psychology, 70*, 137-158. doi:10.1111/bmsp.12078

Straat, J. H., Van der Ark, L. A., & Sijtsma, K. (2016). Using conditional association to identify locally Independent item sets. *Methodology, 12*, 117-123. doi:10.1027/16142241/a000115

Zuccarella-Hackl, C., von Kaenel, R., Thomas, L., Kuebler, P., Schmid, J. P., Mattle. H. P., Mono, M. L., Rieben, R., Wiest, R., & Wirtz, P. H. (2016). Higher macrophage superoxide anion production in coronary artery disease (CAD) patients with Type D personality. *Psychoneuroendocrinology, 68*, 186-193. doi:10.1016/j.psyneuen.2016.02.031

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)
```

ICC *Intraclass correlation*

Description

Computes to intraclass correlation for multilevel data (ICC; Snijders & Bosker, 1999, p. 17) for each item and the total scale of a questionnaire (Koopman et al., in press) and the F-test for the null hypothesis that the (total scale) ICC is zero (Snijders & Bosker, 2012, p. 22)

Usage

```
ICC(X)
```

Arguments

`X`	matrix or data frame of numeric data containing a grouping column and the item scores of `nrow(X)` respondents to `ncol(X) - 1` items. Missing values are not allowed

Value

`itemICC`	The ICC per item
`scaleICC`	The ICC for the total scale and the corresponding results for the F-test

Author(s)

Letty Koopman <`V.E.C.Koopman@uva.nl`> L. Andries van der Ark <`L.A.vanderArk@uva.nl`>

References

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press). A two-step, test-guided Mokken scale analysis for nonclustered and clustered data. *Quality of Life Research*. (advanced online publication) doi:10.1007/s11136021028402

Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel analysis: An introduction to basic and advanced multilevel modeling* (2nd ed.). Sage.

See Also

`SWMDK`, `MLcoefH`,

Examples

```
# Data example (Koopman et al., 2020)
data(SWMDK)

# Compute ICC
mokken::ICC(SWMDK)
```

`mcmi` *Millon Clinical Multiaxial Inventory*

Description

Scores of 1208 patients and inmates on 44 dichotomous items from a Dutch version of the Millon Clinical Multiaxial Inventory-III.

Usage

```
data(mcmi)
```

Format

A 1208 by 44 data frame containing integers.

Details

The data were collected by Gina Rossi (Rossi et al., 2010) as part of a larger project. This subset of 44 items was used to demonstrate diagnostic classification models (cognitive diagnosis models) by de la Torre et al. (2018), Van der Ark et al. (2019), and Sijtsma & Van der Ark (2020). The Q matrix used in the analyses is an attribute. Both the items and the respondents have been deidentified. The deidentified data do allow to replicate the analyses in Sijtsma and Van der Ark (2020, chapter 5), using the code available from `https://osf.io/e9jrz`.

Source

Data were kindly made available by Gina Rossi from the Vrije Universiteit Brussel, Belgium. The original Millon Clinical Multiaxial Inventory-III was developed by Millon (1994).

References

de la Torre, J., Van der Ark, L. A., & Rossi, G. (2018). Analysis of clinical data from cognitive diagnosis modelling framework. *Measurement and Evaluation in Counseling and Development, 51*, 281-296. doi:10.1080/07481756.2017.1327286

Millon, T. (1994). *Manual for the MCMI-III*. Minneapolis, MN: National Computer Systems.

Rossi, G., Elklit, A., & Simonsen, E. (2010). Empirical evidence for a four factor framework of personality disorder organization: Multigroup confirmatory factor analysis of the million clinical multiaxial inventory-III personality disorders scales across Belgian and Danish data samples. *Journal of Personality Disorders, 24*, 128-150. doi:10.1521/pedi.2010.24.1.128

Sijtsma, K., & Van der Ark, L. A. (2020), *Measurement models for psychological attributes.* Chapman and Hall/CRC Press. `https://www.routledge.com/Measurement-Models-for-Psychological-Attributes/Sijtsma-Ark/p/book/9780367424527`

Van der Ark, L. A., Rossi, G., & Sijtsma, K. (2019). Nonparametric item response theory and Mokken scale analysis, with relations to latent class models and cognitive diagnostic models. In M. von Davier & Y.- S. Lee (Eds.), *Handbook of diagnostic classification models* (pp. 21-45). Springer. doi:10.1007/9783030055844

Examples

```
data(mcmi)

# Q matrix
attributes(mcmi)$Q
```

MLcoefH *Two-level scalability coefficents H*

Description

Computes all types of two-level scalability coefficients (Snijders, 2001; Crisan et al., 2016), that is, between- and within-rater item-pair scalability coefficents and their ratio (HBij, HWij, and BWij, respectively), between- and within-rater item scalability coefficients and their ratio (HBi, HWi, and BWi, respectively), and between- and within-rater total scale scalability coefficients and their ratio (HB, HW, and BW, respectively). In addition, standard errors are estimated (Koopman, et al., 2020) and if requested incorporated in confidence intervals (Koopman et al., in press a, in press b). Note that this version is an adaptation of the estimation methods described in Snijders (1999) and in Koopman et al. (2020), as the group proportions are now by default weighted for group size (Koopman, et al. , in press a). As a result, the estimates for the autonomySupport data differs from the printed table in Koopman et al. (2020). The estimated variance-covariance matrix of the coefficients can also be printed if requested.

Usage

```
MLcoefH(X, se = TRUE, ci = FALSE, nice.output = TRUE, subject = 1,
        fixed.itemstep.order = NULL, weigh.props = TRUE,
        type.ci = "WB", cov.mat = FALSE)
```

Arguments

`X`	matrix or data frame of numeric data containing a subject indicator column and the responses of `nrow(X)` raters to `ncol(X) - 1` items. Missing values are not allowed
`se`	Logical: If `TRUE`, the standard errors are printed alongside the scalability coefficients
`ci`	The confidence level between 0 and 1 of the range-preserving confidence intervals. If FALSE (default), no confidence intervals are printed (Koopman et al., in press a).
`nice.output`	Logical: If `TRUE`, scalability coefficients and standard errors are combined in an a single object of class `noquote`. Item-pair ratios BWij are only printed if `FALSE`
`subject`	Represents the subject column. Default is column 1.
`fixed.itemstep.order`	matrix with number of rows equal to the number of item steps (m) and number of columns equal to the number of items (J). The matrix should consis the integers

1 : (m * J), indicating a predefined order of the items steps with respect to popularity. Value 1 indicates the easiest (most popular) item step, value (m * J) indicates the most difficult item step.

weigh.props
: If TRUE: Use weighted proportions across groups to estimate coefficients and standard errors, if FALSE: Use averaged proportions across groups to estimate coefficients and standard errors.

type.ci
: If WB, Wald-based confidence interval are printed, if RP range-preserving confidence intervals are printed (Koopman et al., in press a, in press b). Default is WB. Only used if ci is specified.

cov.mat
: Logical: If TRUE, the variance-covariance matrices of the estimated coefficients are printed. Default is FALSE.

Details

If se = TRUE and nice.output = TRUE, the result is a list of 3 objects of class noquote; if se = TRUE and nice.output = FALSE, the result is a list of 3 matrices, one per set of coefficients; and if se = FALSE, the result is a list of 3 matrices containing only the scalability coefficients; if ci is specified and nice.output = TRUE, there is one additional matrix for the ci's of the Hij coefficients. Computation of standard errors can be slow for a combination of many subjects and a large number of items; if cov.mat = TRUE three additional matrices are printed with the variance-covariances of Hij, Hi, and H.

Value

Hij
: Scalability coefficients of the item pairs, upper triangle are the between-rater coefficients, and the lower triangle the within-rater coefficients (possibly with standard errors and/or confidence intervals; see details). If nice.output = FALSE this returns a matrix with the type of coefficients per column

Hi
: Between- and within-rater scalability coefficients of the items and their ratio (possibly with standard errors and/or confidence intervals; see details)

H
: between- and within-rater scalability coefficients of the entire scale and their ratio (possibly with standard error and/or confidence intervals; see details)

Author(s)

Letty Koopman <V.E.C.Koopman@uva.nl> L. Andries van der Ark <L.A.vanderArk@uva.nl>

References

Crisan, D. R., Van de Pol, J. E., & Van der Ark, L. A. (2016). Scalability coefficients for two-level polytomous item scores: An introduction and an application. In L. A. Van der Ark, D. M. Bolt, W.-C. Wang, & M. Wiberg (Eds.), *Quantitative psychology research*. Springer. doi:10.1007/9783319387598_11

Koopman, L., Zijlstra, B. J. H. & Van der Ark, L. A., (2020). Standard errors of two-lvel scalability coefficients. *British Journal of Statistical and Mathematical Psychology, 73,* 213-236. doi:10.1111/bmsp.12174

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press a). A two-step, test-guided Mokken scale analysis for nonclustered and clustered data. *Quality of Life Research*. (advanced online publication) doi:10.1007/s11136021028402

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press b). Range-preserving confidence intervals and significance tests for scalability coefficients in Mokken scale analysis. In M. Wiberg, D. Molenaar, J. Gonzalez, & Kim, J.-S. (Eds.), *Quantitative Psychology; The 1st Online Meeting of the Psychometric Society, 2020*. Springer. doi:10.1007/9783030747725_16

Snijders, T. A. B. (2001). Two-level non-parametric scaling for dichotomous data. In A. Boomsma, M. A. J. van Duijn, & T. A. B. Snijders (Eds.), *Essays on item response theory* (pp. 319-338). Springer. doi:10.1007/9781461301691_17

See Also

MLcoefZ,coefH,

Examples

```
# Small data example (Koopman et al., 2019)
smallData <- data.frame(Subs = c(1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3),
                        Xa   = c(0, 0, 1, 0, 1, 1, 1, 2, 1, 0, 1, 2, 0, 0, 0),
                        Xb   = c(0, 0, 1, 0, 2, 2, 2, 1, 2, 1, 2, 2, 1, 1, 0),
                        Xc   = c(1, 0, 0, 0, 1, 1, 2, 1, 2, 0, 1, 1, 2, 1, 0))

MLcoefH(smallData)

# Compute also the range-preserving confidence intervals
MLcoefH(smallData, ci = .95)

# Print variance-covariance matrices
MLcoefH(smallData, cov.mat = TRUE)

# Load real data example. Note that due to an estimation adaptation (Koopman et al., 2020)
# the results differ from the table in Koopman et al. (2019).
data(autonomySupport)

# Compute scalability coefficients with or without standard errors, range-preserving
# confidence intervals, nice output

 H.se.nice <- MLcoefH(autonomySupport)
 H.se.nice


 H.se.not  <- MLcoefH(autonomySupport, nice.output = FALSE)
 H.se.not


 H.se.ci.nice <- MLcoefH(autonomySupport, ci = .95)
 H.se.ci.nice


 H.se.ci.not <- MLcoefH(autonomySupport, ci = .95, nice.output = FALSE)
```

```
H.se.ci.not


H.nice    <- MLcoefH(autonomySupport, se = FALSE)
H.nice


H.not     <- MLcoefH(autonomySupport, se = FALSE, nice.output = FALSE)
H.not
```

MLcoefZ *Computation of Z-Values for two-level scalability coefficients*

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 (within- and between-rater versions), are significantly greater a specified lowerbound using the delta method (Koopman et al., in press a). The test uses either Wald-based (`WB`) or range-preserving (`RP`) asymptotic theory (Koopman et al., in press b).

Usage

```
MLcoefZ(X, lowerbound = 0, type.z = "WB")
```

Arguments

`X`	matrix or data frame of numeric data containing the responses of `nrow(X)` respondents to `ncol(X) - 1` items. The first column of `X` is assumed to be a subject column, see `?MLcoefH()` for details. Missing values are not allowed
`lowerbound`	Value of the null hypothesis to which the scalability are compared to compute the z-score (see details), 0 <= `lowerbound` < 1. The default is 0.
`type.z`	Indicates which type of z-score is computed: "WB": Wald-based z-score based on standard errors as approximated by the delta method (Kuijpers et al., 2013; Koopman et al., in press a); "RP": Range-preserving z-score, also based on the delta method (Koopman et al., in press b). The default is "WB".

Details

For the estimated item-pair coefficient Hij with standard error $SE(Hij)$, the Z-score is computed as

$$Zij = (Hij - lowerbound)/SE(Hij)$$

if `type.z = "WB"`, and the Z-score is computed as

$$Zij = -(log(1 - Hij) - log(1 - lowerbound))/(SE(Hij)/(1 - Hij))$$

if `type.z = "RP"` (Koopman et al., in press b). For the estimate item-scalability coefficients Hi and total-scalbility coefficients H a similar procedure is used. Standard errors of the Z-scores 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`> L. Koopman

References

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press a). A two-step, test-guided Mokken scale analysis for nonclustered and clustered data. *Quality of Life Research*. (advanced online publication) doi:10.1007/s11136021028402

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press b). Range-preserving confidence intervals and significance tests for scalability coefficients in Mokken scale analysis. In M. Wiberg, D. Molenaar, J. Gonzalez, & Kim, J.-S. (Eds.), *Quantitative Psychology; The 1st Online Meeting of the Psychometric Society, 2020*. Springer. doi:10.1007/9783030747725_16

See Also

`MLcoefH`

Examples

```
data(SWMD)

# Compute the Z-score using lowerbound 0
MLcoefZ(SWMD)

# Using lowerbound .1
MLcoefZ(SWMD, lowerbound = .1)
```

`MLweight` *Weights for Guttman Errors in two-level test data*

Description

Computes weights for Guttman errors in two-level test data (Koopman et al., 2017)

Usage

```
MLweight(X, maxx = NULL, minx = NULL, itemstep.order = NULL)
```

Arguments

X	Data matrix with a subject column and two item columns. The subject column is assumed to be the first.
maxx	The highest possible answer category. If not specified it is determined by using the highest item score.
minx	The lowest possible answer category. If not specified it is determined by using the lowest item score.
itemstep.order	The two columns pertaining the two items in question, from a (possibly larger) matrix with number of rows equal to the number of item steps (m) and number of columns equal to the number of items (J). The matrix should consis the integers 1 : (m * J), indicating a predefined order of the items steps with respect to popularity. Value 1 indicates the easiest (most popular) item step, value (m * J) indicates the most difficult item step.

Value

Returns a vector with the weights for each item-score pattern of a given item-pair. In case of ties in item popularities the average weights across possible item-orderings are returned.

Author(s)

L. Koopman <V.E.C.Koopman@uva.nl> L. A. van der Ark <L.A.vanderArk@uva.nl>

References

Koopman, L., Van der Ark, L. A., & Zijlstra, B. J. H. (2017). *Weighted Guttman Errors: Handling Ties and Two-Level Data*. In L. A. Van der Ark, S. Culpepper, J. A. Douglas, W.-C. Wang, & M. Wiberg (Eds.), *Quantitative Psychology: The 81st Annual Meeting of the Psychometric Society, Asheville, North Carolina, 2016* (pp. 183-190). Springer. doi:10.1007/9783319562940_17

See Also

MLcoefH

Examples

```
smallData <- data.frame(Subs = c(1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3),
                        Xa = c(2, 0, 0, 1, 0, 2, 2, 0, 2, 2, 1, 2, 1, 2, 2),
                        Xb = c(1, 1, 1, 0, 1, 2, 2, 1, 2, 2, 1, 0, 2, 2, 2),
                        Xc = c(0, 0, 0, 1, 0, 2, 2, 1, 2, 1, 0, 0, 1, 1, 2))

# Compute the weights
I <- 3
for(i in 1:(I - 1)) for(j in (i + 1):I)
print(MLweight(smallData[, c(1, i + 1, j + 1)], minx = 0, maxx = 2))
```

`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

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023a). Assumptions and Properties of Two-Level Nonparametric Item Response Theory Models. Manuscript submitted for publication.

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023b). Evaluating Model Fit in Two-Level Mokken Scale Analysis. Manuscript submitted for publication.

Ligtvoet, R., L. A. van der Ark, J. M. te Marvelde, & K. Sijtsma (2010). Investigating an invariant item ordering for polytomously scored items. *Educational and Psychological Measurement, 70*, 578-595. doi:10.1177/0013164409355697

Ligtvoet, R., L. A. van der Ark, W. P. Bergsma, & K. Sijtsma (2011). Polytomous latent scales for the investigation of the ordering of items. *Psychometrika, 76*, 200-216. doi:10.1007/s11336010-91998

Sijtsma, K., R. R. Meijer, & Van der Ark, L. A. (2011). Mokken scale analysis as time goes by: An update for scaling practitioners. *Personality and Individual Differences, 50*, 31-37. doi:10.1016/j.paid.2010.08.016

Van der Ark, L. A. (2012). New developjements in Mokken scale analysis in **R**. *Journal of Statistical Software, 48* (5), 1-27. doi:10.18637/jss.v048.i05

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, & C. M. Woods (Eds.), *New developments in quantitative psychology* (pp. 59-68). New York: Springer. doi:10.1007/97814614-93488_5

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)

# Compute two-level fit statistics (Koopman et al., 2023a, 2023b)
data("autonomySupport")
dat <- autonomySupport[, -1]
groups <- autonomySupport[, 1]
autonomyMIIO <- check.iio(dat, item.selection = FALSE, level.two.var = groups)
summary(autonomyMIIO)
plot(autonomyMIIO)
```

`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

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023a). Assumptions and Properties of Two-Level Nonparametric Item Response Theory Models. Manuscript submitted for publication.

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023b). Evaluating Model Fit in Two-Level Mokken Scale Analysis. Manuscript submitted for publication.

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

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v020.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, & C. M. Woods (Eds.), *New developments in quantitative psychology* (pp. 59-68). New York: Springer. doi:10.1007/97814614-93488_5

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)

# Compute two-level fit statistics (Koopman et al., 2023a, 2023b)
data("autonomySupport")
dat <- autonomySupport[, -1]
groups <- autonomySupport[, 1]
autonomyMM <- check.monotonicity(dat, level.two.var = groups)
summary(autonomyMM)
plot(autonomyMM)
```

---

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., & Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. IEC ProGAMMA.

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v020.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., & Sijtsma, K. (2000) *User's Manual MSP5 for Windows* [Software manual]. IEC ProGAMMA.

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v020.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, & C. M. Woods (Eds.), *New developments in quantitative psychology* (pp. 59-68). New York: Springer. doi:10.1007/97814614-93488_5

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 intvolved; `#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 & 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

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023a). Assumptions and Properties of Two-Level Nonparametric Item Response Theory Models. Manuscript submitted for publication.

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023b). Evaluating Model Fit in Two-Level Mokken Scale Analysis. Manuscript submitted for publication.

Ligtvoet, R., L. A. van der Ark, J. M. te Marvelde, & K. Sijtsma (2010). Investigating an invariant item ordering for polytomously scored items. *Educational and Psychological Measurement, 70*, 578-595. doi:10.1177/0013164409355697

Ligtvoet, R., L. A. van der Ark, W. P. Bergsma, & K. Sijtsma (2011). Polytomous latent scales for the investigation of the ordering of items. *Psychometrika, 76*, 200-216. doi:10.1007/s11336010-91998

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

Sijtsma, K., R. R. Meijer, & Van der Ark, L. A. (2011). Mokken scale analysis as time goes by: An update for scaling practitioners. *Personality and Individual Differences, 50*, 31-37. doi:10.1016/j.paid.2010.08.016

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software, 20* (11), 1-19. doi:10.18637/jss.v020.i11

Van der Ark, L. A. (2012). New developments in Mokken scale analysis in R. *Journal of Statistical Software, 48*(5), 1-27. doi:10.18637/jss.v048.i05

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)

# Compute two-level fit statistics (Koopman et al., 2023a, 2023b)
data("autonomySupport")
dat <- autonomySupport[, -1]
groups <- autonomySupport[, 1]
autonomyMIIO <- check.iio(dat, item.selection = FALSE, level.two.var = groups)
summary(autonomyMIIO)
```

`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 & Sijtsma, 2000, pp. 49, 74).

Author(s)

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

References

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023a). Assumptions and Properties of Two-Level Nonparametric Item Response Theory Models. Manuscript submitted for publication.

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (2023b). Evaluating Model Fit in Two-Level Mokken Scale Analysis. Manuscript submitted for publication.

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

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

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v020.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)

# Compute two-level fit statistics (Koopman et al., 2023a, 2023b)
data("autonomySupport")
dat <- autonomySupport[, -1]
groups <- autonomySupport[, 1]
autonomyMM <- check.monotonicity(dat, level.two.var = groups)
summary(autonomyMM)
```

`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 & 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*. De Gruyter.

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

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v020.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 & 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*. De Gruyter.

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

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. Sage.

Van der Ark, L. A. (2007). Mokken scale analysis in **R**. *Journal of Statistical Software*. doi:10.18637/jss.v020.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)
```

SWMD *SWMD Data Subset*

Description

A subset of the COOL5-18 data (see below) with scores of 651 pupils nested in 30 classes on the 7-item Schaal Welbevinden Met Docenten [Scale Well-Being With Teachers] (Peetsma, et al., 2001; Zijsling, et al., 2017). See `attributes(SWMD)` for the original item labels and content. R-code to get this subsample is available as online supplement to Koopman et al. (in press a).

Usage

```
data(SWMD)
```

Format

A 651 by 8 data frame containing integers. The first column reflects a classroom indicator, the remaining columns the 7 items, see `colnames(SWMD)`.

Details

Each item has five ordered answer categories from *not true at all* (score 0) to *completely true* (score 4).

Item	*Short*	*Content*
Item1	lv_wdo1	The teachers usually know how I feel
Item2	lv_wdo2	I can talk about problems with the teachers
Item3	lv_wdo3	If I feel unhappy, I can talk to the teachers about it
Item4	lv_wdo4	I feel at ease with the teachers
Item5	lv_wdo5	The teachers understand me
Item6	lv_wdo6	I have good contact with the teachers
Item7	lv_wdo7	I would prefer to have other teachers (inversely coded)

The items were translated from Dutch. For the original items, see p. 79 in Zijsling et al. (2017). The scores on these items plus seven additional items are available in dataset `SWMDK`.

Source

Data is a subset of respondents and items from the large-scale cohort study COOL5-18. `http://www.cool5-18.nl/` (Zijsling et al., 2017). For entire dataset see doi:10.17026/danszfpegnq Dataset is accessible after login.

References

Koopman, L. Zijlstra, B. J. H, & Van der Ark, L. A. (2020). *A two-step procedure for scaling multilevel data using Mokken's scalability coefficients*. Manuscript submitted for publication.

Peetsma, T. T. D., Wagenaar, E., & De Kat, E. (2001). School motivation, future time perspective and well-being of high school students in segregated and integrated schools in the Netherlands and the role of ethnic self-description. In J. Koppen, I. Lunt, & C. Wulf (Eds.), *Education in Europe. Cultures, Values, Institutions in Transition* (pp. 54-74). Waxmann.

Zijsling, D., Keuning, J., Keizer-Mittelhaeuser, M.-A., Naaijer, H., & Timmermans, A. (2017). *Cohortonderzoek COOL5-18: Technisch rapport meting VO-3 in 2014*. Onderwijs/Onderzoek.

See Also

MLcoefH, ICC,

Examples

```
# Data example (Koopman et al., 2020)
data(SWMD)

# Item content, see labels
attributes(SWMD)$labels

# Compute ICC
ICC(SWMD)
```

SWMDK *SWMDK Data Subset*

Description

A subset of the COOL5-18 data (see below) with scores of 639 pupils nested in 30 classes on the 7-item Schaal Welbevinden Met Docenten [Scale Well-Being With Teachers] and 6-item Schaal Welbevinden met Klasgenoten [Scale Well-Being With Classmates] (Peetsma et al., 2001; Zijsling et al., 2017). See `attributes(SWMDK)` for the original item labels and content. R-code to get this subsample is available as online supplement to Koopman et al. (in press a).

Usage

```
data(SWMDK)
```

Format

A 639 by 14 data frame containing integers. The first column reflects a classroom indicator, the remaining columns the 13 items, see `colnames(SWMDK)`.

Details

Each item has five ordered answer categories from *not true at all* (score 0) to *completely true* (score 4).

Item	*Short*	*Content*
Item1	lv_wdo1	The teachers usually know how I feel
Item2	lv_wdo2	I can talk about problems with the teachers
Item3	lv_wdo3	If I feel unhappy, I can talk to the teachers about it
Item4	lv_wdo4	I feel at ease with the teachers
Item5	lv_wdo5	The teachers understand me
Item6	lv_wdo6	I have good contact with the teachers
Item7	lv_wdo7	I would prefer to have other teachers (inversely coded)
Item8	lv_wkl1	I have a lot of contact with my classmates
Item9	lv_wkl2	I would prefer to be in a different class (inversely coded)
Item10	lv_wkl3	We have a nice class
Item11	lv_wkl4	I get along well with my classmates
Item12	lv_wkl5	I sometimes feel alone in the class (inversely coded)
Item13	lv_wkl6	I enjoy hanging out with my classmates

The items were translated from Dutch. For the original items, see pp. 79-83 in Zijsling et al. (2017). The first seven items are also available in dataset `SWMD`.

Source

Data is a subset of respondents and items from the large-scale cohort study COOL5-18. `http://www.cool5-18.nl/` (Zijsling et al., 2017). For entire dataset see doi:10.17026/danszfpegnq Dataset is accessible after login.

References

Koopman, L., Zijlstra, B. J. H., & Van der Ark, L. A. (in press a). A two-step, test-guided Mokken scale analysis for nonclustered and clustered data. *Quality of Life Research*. (advanced online publication) doi:10.1007/s11136021028402

Peetsma, T. T. D., Wagenaar, E., & De Kat, E. (2001). School motivation, future time perspective and well-being of high school students in segregated and integrated schools in the Netherlands and the role of ethnic self-description. In J. Koppen, I. Lunt, & C. Wulf (Eds.), *Education in Europe. Cultures, Values, Institutions in Transition* (pp. 54-74). Waxmann.

Zijsling, D., Keuning, J., Keizer-Mittelhaeuser, M.-A., Naaijer, H., & Timmermans, A. (2017). *Cohortonderzoek COOL5-18: Technisch rapport meting VO-3 in 2014*. Onderwijs/Onderzoek.

See Also

`coefH`, `MLcoefH`, `ICC`, `SWMD`

Examples

```
# Data example (Koopman et al., in press)
data(SWMDK)

# Item content, see labels
attributes(SWMDK)$labels

# Compute ICC
ICC(SWMDK)
```

`transreas` *Transitive Reasoning*

Description

Data came from 12 dichtomous items administered to 425 children in grades 2 through 6 (Verweij, Sijtsma, & 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 & 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., & Koops, W. (1996). A Mokken scale for transitive reasoning suited for longitudinal research. *International Journal of Behavioral Development, 23*, 241-264. doi:10.1177/016502549601900115

Sijtsma, K., & Molenaar, I. W. (2002) *Introduction to nonparametric item response theory*. 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
```

`transreas2` *Transitive Reasoning Data*

Description

Scores of 606 school children on 16 dichotomous transitive reasoning items.

Usage

```
data(transreas2)
```

Format

A 606 by 16 data frame containing integers.

Details

The data were collected by Samantha Bouwmeester (Bouwmeester & Sijtsma, 2004). The design of the items is included as attributes. The respondents have been deidentified. The deidentified data do allow to replicate the analyses in Sijtsma and Van der Ark (2020, chapter 4), using the code available from `https://osf.io/e9jrz`. Note that the package `mokken` includes another transitive reasoning data set.

Source

Data were kindly made available by Samantha Bouwmeester from the Erasmus Universiteit Rotterdam, The Netherlands.

References

Bouwmeester, S., & Sijtsma, K. (2004). Measuring the ability of transitive reasoning, using product and strategy information. *Psychometrika, 69*, 123-146. doi:10.1007/BF02295843

Sijtsma, K., & Van der Ark, L. A. (2020), *Measurement models for psychological attributes.* Chapman and Hall/CRC Press. https://www.routledge.com/Measurement-Models-for-Psychological-Attributes/Sijtsma-Ark/p/book/9780367424527

Examples

```
data(transreas2)

# Create Table 4.2 from Sijtsma & Van der Ark (2020)
data.frame (Item = 1: 16,
            taskContent = attr(transreas2, which = "taskContent"),
            presentationForm = attr(transreas2, which = "presentationForm"),
            taskFormat = attr(transreas2, which = "taskFormat"),
            pValues = round(apply(transreas2, 2, mean), 2))
```

---

trog *trog Data*

Description

A clustered-item dataset with scores of 210 children (measurement taken at a first time point of a larger developmental study; Brinchmann et al., 2019) that took the Norwegian adaptation of the Test for Reception of Grammar (TROG; Bishop, 1979). The TROG consists of C=20 clusters of Jc=4 dichotomously scored items for all c. and all kids were administered each of the J=80 items. This item set is used as a real data example to demonstrate how to investigate the ordering structure of a test in Koopman & Braeken (2024).

Usage

```
data(autonomySupport)
```

Format

A 210 by 80 data frame containing dichotomous data. Each column reflects an item, each row a respondent. The column labels reflect the cluster (letters) and item within the cluster (numbers). For example, item a1 is the first item in cluster a, see `colnames(trog)`.

Details

Each item is dichotomously scored, where 1 reflects a correct response and 0 an incorrect response.

References

Bishop, D. V. M. (1979). Comprehension in developmental language disorders. *Developmental Medicine & Child Neurology, 21(2)*, 225-238. doi:10.1111/j.14698749.1979.tb01605.x.

Brinchmann, E. I., Braeken, J., & Lyster, S.-A. H. (2019). Is there a direct relation between the development of vocabulary and grammar? *Developmental Science, 22(1)*, 1-13. doi:10.1111/desc.12709.

Koopman, L. & Braeken, J. (2024). Investigating the Ordering Structure of Clustered Items Using Nonparametric Item Response Theory. Manuscript submitted for publication.

See Also

`check.iio`,

Examples

```
data(trog)
```

`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., & 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. doi:10.1207/S15327906MBR3503_03

Sijtsma, K., & Van der Ark, L. A. (2003). Investigation and treatment of missing item scores in test and questionnaire data. *Multivariate Behavioral Research, 38*, 505-528. doi:10.1207/s15327906mbr3804_4

Van Ginkel, J. R., Van dec Ark, L. A., & Sijtsma, K. (2007). Multiple imputation of item scores in test and questionnaire data, and influence on psychometric results. *Multivariate aBehavioral Research, 42*, 387-414. doi:10.1080/00273170701360803

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