Package 'pairwise'

April 7, 2014

Type Package

Maintainer Joerg-Henrik Heine <jhheine@googlemail.com>

Date 2014-04-04

Author Joerg-Henrik Heine <jhheine@googlemail.com>

Version 0.2.2

Encoding UTF-8

License GPL-3

Depends R (>= 2.10.1)

Title Rasch Model Parameters by Pairwise Algorithm

Description The package pairwise offers functions for the explicit calculation -- not estimation! -- of the rasch item parameters for dichotomous and polytomous item responses, using a pairwise comparison approach.

Suggests PP

Collate 'pairwise-package.r' 'make.incidenz.R' 'pair.R' 'pers.R' 'i.ptb.R' 'i.dummy.R' 'i.polyptb.R' 'ptbis.R' 'plot.pair.R' 'summary.pair.R' 'plot.pairSE.R' 'pairSE.R' 'summary.pairSE.R' 'grm.R' 'plot.grm.R' 'summary.grm.R' 'ftab.R' 'catprob.R' 'i.pvx.R' 'summary.pers.R' 'esc.R' 'i.pvx.matrix.R' 'i.str.pattern.R' 'i.expscore.R' 'pairwise.item.fit.R' 'pairwise.person.fit.R' 'i.GewLL.R' 'i.PersPar.R' 'i.dataprep1.R' 'i.missing_group.R' 'gif.R'

R topics documented:

bfiN 2
bfiN_miss 3
catprob 4
cog 5
cogBOOKLET 5
DEU_PISA2012 6
esc 7
ftab 8
gif 8

grm . . . 9
make.incidenz . . . 11
pair . . . 13
pairSE . . . 14
pairwise . . . 16
pairwise.item.fit . . . 17
pairwise.person.fit . . . 18
pers . . . 19
plot.grm . . . 20
plot.pair . . . 21
plot.pairSE . . . 22
ptbis . . . 23
sim200x3 . . . 23
summary.pers . . . 24

Index **25**

`bfiN` *5 polytomous personality items*

Description

Data from 2800 subjects answering to 5 neuroticism items with 6 answer categories (0-5) of the bfi dataset originally included in the R-package `{psych}`.

Usage

```
data(bfiN)
```

Format

A data frame containing 5 variables and 2800 obsevations.

Details

The other variables from the original bfi dataset were skipped to have a simple example data frame. For further Information on the original dataset see R-package `{psych}`.

Source

http://cran.r-project.org/web/packages/psych/index.html

References

Revelle, William (2012), psych: Procedures for Psychological, Psychometric, and Personality Research.*R package version 1.2.12*

Examples

```
data(bfiN)
dim(bfiN)
#################################################################
names(bfiN) # show all variable names of data.frame bfiN
range(bfiN,na.rm=TRUE) # checking the valid response range
```

`bfiN_miss` *5 polytomous personality items*

Description

Data from 2800 subjects answering to 5 neuroticism items with 6 answer categories (0-5) of the bfi dataset originally included in the R-package {psych} with artificial missing data (see details) .

Usage

```
data(bfiN_miss)
```

Format

A data frame containing 5 variables and 2800 obsevations.

Details

This dataset is the same like the dataset {bfiN} included in this package, exept for the amount of missing data, which were additional created in that way, having aprox. 15% missing for each of the 5 variables by random.

The other variables from the original bfi dataset were skipped to have a simple example data frame. For further Information on the original dataset see R-package {psych}.

Source

http://cran.r-project.org/web/packages/psych/index.html

References

Revelle, William (2012), psych: Procedures for Psychological, Psychometric, and Personality Research.*R package version 1.2.12*

Examples

```
data(bfiN_miss)
dim(bfiN_miss)
#################################################################
names(bfiN_miss) # show all variable names of data.frame bfiN_miss
range(bfiN_miss,na.rm=TRUE) # checking the valid response range
colSums(is.na(bfiN_miss))/dim(bfiN_miss)[1] # percentage of missing per variable
```

catprob *Category Probability Plots*

Description

plotting function for plotting categoy probability curves

Usage

```
catprob(pair_obj, itemnumber = 1, ra = 4, plot = TRUE,
  ...)
```

Arguments

pair_obj	an object of class "pair" as a result from function pair.
itemnumber	an integer, defining the number of the item to plot the respective categoy probability for. This is set to an arbitrary default value of itemnumber = 1 to avoid error messages when you forget to choose an item to plot the expected score curves for.
ra	an integer, defining the (logit) range for x-axis
plot	a logical (default plot = TRUE), defining wether to supress plotting an just return a matrix of category probabilities
...	arguments passed to plot

Details

to come ...

Value

a plot or a matrix with category probabilities.

Examples

```
########
data(sim200x3)
result <- pair(sim200x3)
catprob(pair_obj = result, itemnumber = 2 )
data(bfiN)
result <- pair(bfiN)
catprob(pair_obj = result, itemnumber = 3 )
```

cog *Math PISA (2003) data*

Description

Data from the german sample of the PISA 2003 survey, containing 31 dichotomous items from the math task.

Usage

```
data(cog)
```

Format

A data frame containing 34 variables and 4660 obsevations.

Details

The first 3 variables are ID variables. For further Information on variables and their meaning see the codebook PDF file available at http://pisa2003.acer.edu.au/downloads.php

Source

http://pisa2003.acer.edu.au/downloads.php

References

Database - PISA 2003, *Downloadable Data*, http://pisa2003.acer.edu.au/downloads.php

Examples

```
data(cog)
dim(cog)
############################################################
names(cog) # show all variable names of data.frame cog
names(cog[,4:34]) # show the variable names of the math items
names(cog[,1:3]) # show the variable names of the ID variables
```

cogBOOKLET *Booklet allocation table for Math PISA (2003) data*

Description

a `data.frame` containing a booklet allocation table for the cognitive Data `cog` in this packge, which holds 31 dichotomous items from the math task from the german sample of the PISA 2003 survey.

Usage

```
data(cogBOOKLET)
```

Format

A `data.frame` containing 31 rows.

Details

For further Information on variables and their meaning see the codebook PDF file available at http://pisa2003.acer.edu.au/downloads.php

Source

http://pisa2003.acer.edu.au/downloads.php

References

Database - PISA 2003, *Downloadable Data*, http://pisa2003.acer.edu.au/downloads.php

Examples

```
data(cogBOOKLET)
cogBOOKLET
```

DEU_PISA2012 *Data from PISA 2012 - German Sample*

Description

Selectetd data for 5001 'subjects' who participated in the PISA 2012 survey.

Usage

```
data(DEU_PISA2012)
```

Format

A list containing ... .

Details

The data is based on freely down loadable data on the official OECD page - see source. The general structure of the data in list format, is described in an PDF document available in the User guides, package vignettes and other documentation section.

Source

http://pisa2012.acer.edu.au/downloads.php

References

To come ...

Examples

```
#################################################################
data(DEU_PISA2012)
str(DEU_PISA2012)
```

---

esc *Expected Score Curves Plots*

Description

plotting function for plotting expected score curves

Usage

```
esc(pers_obj, itemnumber = 1, integ = 6, ra = 4,
  nodes = 100, lwd = 2, ...)
```

Arguments

pers_obj	an object of class "pers" as a result from function pers.
itemnumber	an integer, defining the number of the item to plot the respective categoy probability for. This is set to an arbitrary default value of itemnumber = 1 to avoid error messages when you forget to choose an item to plot the expected score curves for.
integ	either an integer defining the number of (ability) groups to integrate the empirical theta vector or the character expression "all" to plot the empirical theta distribution at the respective item score using symbols (see example).
ra	an integer, defining the (logit) range for x-axis
nodes	numer of integration nodes
lwd	see plot
...	arguments passed to plot

Details

to come ...

Examples

```
########
data(bfiN)
result <- pers(pair(bfiN))
esc(pers_obj=result,1,lwd=2) # plot for first item
esc(pers_obj=result,2,lwd=2) # plot for second item
for(i in 1:5){esc(pers_obj=result,i,lwd=2)}
#########
esc(pers_obj=result,2,integ="all",lwd=2) # plot for secod item
```

ftab *Tabulating Answer Categories in Data*

Description

function tabulating (answer) categories in X.

Usage

```
ftab(X, catgories = NULL, na.omit = FALSE)
```

Arguments

X	Data as a "matrix", a "data.frame" or even a "vector" or "factor". "vector" or "factor" are coerced to a "data.frame" with one column.
catgories	optional a vector ("numeric" or "character") containig the categories to tabulate. At default (catgories=NULL) the fuction looks for unique categories in X.
na.omit	logical (default: na.omit=FALSE) wether to return frequencies for missing values, NAs.

Details

X can either be a ("numeric" or "character") "matrix" containing response vectors of persons (rows) or a "data.frame" containing "numeric", "character" or "factor" variables (columns).

Value

a "matrix" with category frequencies

Examples

```
########
data(bfiN)
ftab(bfiN)
data(sim200x3)
ftab(sim200x3)
```

gif *Graphical Item Fit Plots*

Description

plotting function for plotting empirical and model derived category probability curves.

Usage

```
gif(pers_obj, itemnumber = 1, ra = 4, integ = "raw",
  kat = "all", ...)
```

Arguments

pers_obj	an object of class "pers" as a result from function pers.
itemnumber	an integer, defining the number of the item to plot the respective categoy probability for. This is set to an arbitrary default value of itemnumber = 1 to avoid error messages when you forget to choose an item to plot the expected score curves for.
ra	an integer, defining the (logit) range for x-axis
integ	either an integer, defining the number of integration points along the (logit) range on the x-axis to integrate the empirical theta values, or the character expression "raw" (default) which will use the rawscore groups as integration points.
kat	either an integer, defining for which category the empirical category probabilities should be plotted over the model derived category probability curves, or the character expression "all" (default) which will plot the empirical category probabilities for all categories.
...	arguments passed to plot

Details

to come ...

Value

a plot with category probabilities.

Examples

```
########
data(bfiN)
pers_obj <- pers(pair(bfiN))
#### plot empirical category probabilities
gif(pers_obj = pers_obj, itemnumber = 1 )
gif(pers_obj = pers_obj, itemnumber = 1 , integ=8) # integration over 8 points
gif(pers_obj = pers_obj, itemnumber = 1 , integ=8, kat=1) # only for category number 1
```

---

grm *Graphical Model Check*

Description

This function makes the basic calculations for the graphical model check for dicho- or polytomous itemresponse formats. It is more or less a wraper function, internaly calling the function pairSE. Several splitting options are available (see arguments).

Usage

```
grm(daten, m = NULL, teil = "random", splitseed = "no",
  verbose = TRUE, ...)
```

Arguments

daten	daten a data matrix with optionaly named colums (names of items) or a data.frame, potentially with missing values, comprising polytomous responses of respondents (rows) on some items (colums) coded starting with 0 for lowest category to *m*-1 for highest category, with *m* beeing the number of categories for all items.
m	number of response categories for all items - by default *m* is defined as `m = max(daten,na.rm=TRUE)+1`.
teil	Specifies the splitting criterion. Basicly there are three different options available - each with several modes - which are controlled by passing the corresponding character expression to the argument. 1) Using the rawscore for splitting into subsamples with the following modes: `teil = "median"` median raw score split - high score group and low score group; `teil = "mean"` mean raw score split - high score group and low score group; `teil = "score"` splitting `daten` into as many subsamples as there are raw score groups (discarding min and max score group) 2) Dividing the persons in `daten` into subsamples with equal size by random allocation with the following modes: `teil = "random"` (which is equivalent to `teil = "random.2"`) divides persons into two subsamples with equal size. In general the number of desired subsamples must be expressed after the dot in the character expression - e.g. `teil = "random.6"` divides persons into 6 subsamples (with equal size) by random allocation etc. 3) The third option is using a manifest variable as a splitting criterion. In this case a numeric indicating the column number of the variable in `daten` must be passed to the argument - e.g. `teil = 1` indicates that the variable in the first column of `daten` will be used as splitting criterion - (this variable will of course be used only as splitting criterion). The variable in `daten` should be coded as `factor` or a numeric integer vector with min = 1 if `daten` is a matrix.
splitseed	numeric, used for `set.seed(splitseed)` for random splitting - see argument `teil`
verbose	logical, if verbose = TRUE (default) a message about subsampling whe calculation standrderrors is sent to console
...	additional arguments `nsample`, `size`, `seed`, `pot` for caling `pairSE` are passed through - see description for `pairSE`.

Details

The data is splitted in two or more subsamples and then item thresholds, the parameter (Sigma) and their standard errors (SE) for the items according the PCM are calculatetd for each subsample. Additional arguments (see description of function `pairSE`) for parameter calculation are passed through.

WARNING: When using data based on booklet designs with systematically missing values (by design) you have to ensure that in each of the booklet the maximum raw value to reach is equal while using the raw value as splitting criterion.

Value

A (list) object of class `"grm"` containing the item difficulty parameter sigma and their standard errors for two or more subsamples.

A Note on Standard Errors

Estimation of standard errors is done by repeated calculation of item parameters for subsamples of the given data. This procedure is mainly controlled by the arguments `nsample` and `size` (see arguments). With regard to calculation time, the argument `nsample` is the 'time killer'. On the other hand, things (estimation of standard errors) will not necessarily get better when choosing large values for `nsample`. For example choosing `nsample=400` will only result in minimal change for standard error estimation in comparison to (`nsample=30`) which is the default setting (see examples).

References

description of function `pairSE{pairwise}`.

Examples

```
data(bfiN) # loading example data set

# calculating itemparameters and SE for two random allocated subsamples
grm<-grm(daten=bfiN, teil = "random")

summary(grm)

# some examples for plotting options
# plotting item difficulties for two subsamples against each other
# with elipses for a CI = 95% .
plot(grm)

# using triangles as plotting pattern
plot(grm,pch=2)

#plotting without CI ellipses
plot(grm,ci=0,pch=2)

# plotting with item names
plot(grm,itemNames=TRUE)

# Changing the size of the item names
plot(grm,itemNames=TRUE, cex.names = 1.3)

# Changing the color of the CI ellipses
plot(grm,itemNames=TRUE, cex.names = .8, col.error="green")

###### example from details section 'Some Notes on Standard Errors' ########
grm<-grm(daten=bfiN, teil = "random",splitseed=13)
plot(grm)
######
grm_400<-grm(daten=bfiN, teil = "random", splitseed=13 ,nsample=400)
plot(grm_400)
```

`make.incidenz` *Converting a booklet allocation table into a incidenz matrix*

Description

This function converts a booklet allocation table (like in cogBOOKLET) into a incidenz matrix used in the function pers.

Usage

```
make.incidenz(tab, bookid, item_order = NULL,
  info = FALSE)
```

Arguments

tab	a booklet allocation table as a data.frame. The first column is assumed to contain the item names as a character vector (not a factor!) the other columns must be integer vectors containing the information in which booklet(s) the respective item is allocated.
bookid	a integer vector with the same length as the number of persons in the response data giving the information which booklet was assigned to each person.
item_order	optional a character vector with the item names in the order of the itmes in the response data (from first to last column in the response data). By default it is assumend that the item order in the booklet allocation table is already the same as in the response data.
info	logical default: info=FALSE to return just the incidenz matrix. If set to info=TRUE more detailed information about the booklet design ist returned.

Details

It is assumed that there is an equal replicate factor for each item used when constructing the bookletdesign - so every items occures with the same frequency over al booklets of the entire set of booklets.

Value

an incidenz matrix as an object of class "matrix" with 0,1 coding or a "list" with detailed information.

Examples

```
#########################
data(cog);data(cogBOOKLET) # loading reponse and allocation data
table(cog$BOOKID)# show n persons per booklet
names(table(c(as.matrix(cogBOOKLET[,2:5])))) # show booklets in allocation data
d<-(cog[cog$BOOKID!=14,]) # skip persons which got booklet No.14.
inc<-make.incidenz(tab=cogBOOKLET, bookid=d$BOOKID) # make just the incidenz matrix
inc
make.incidenz(tab=cogBOOKLET, bookid=d$BOOKID, info=TRUE) # get some info too
# in this case not necessary but just to show
# using the (item) names in cog to secure the item order in incidenz matrix:
make.incidenz(tab=cogBOOKLET, bookid=d$BOOKID, item_order=names(cog)[4:34])
#######################
```

`pair` *Rasch Item Parameter (Main Function)*

Description

This is the (new) main function for calculation of the item parameter for the dichotomous Rasch Model (Rasch, 1960) and its extension for polytomous items (thurstonian thresholds) according to the Partial Credit Model (Masters, 1982), using a generalization of the pairwise comparison algorithm (Choppin, 1968, 1985; Wright & Masters, 1982). The number of (response) categories may vary accross Items. Missing values up to an high amount in data are allowed, as long as items are proper linked together.

Usage

```
pair(daten, m = NULL, pot = TRUE, zerocor = TRUE,
  ccf = FALSE, ...)
```

Arguments

`daten`	a `data.frame` or `matrix` with optionaly named colums (names of items), potentially with missing values, comprising polytomous or dichotomous (or mixted category numbers) responses of n respondents (rows) on k items (colums) coded starting with 0 for lowest category to *m*-1 for highest category, with *m* beeing a vector (with length k) with the number of categories for the respective item.
`m`	an integer (will be recycled to a vector of length k) or a vector giving the number of response categories for all items - by default (`m = NULL`), m is calculatetd from data, assuming that every response category is at least once present in data. For *'sparse' data* it is *strongly recomended* to explicitly *define the number of categories* by defining this argument.
`pot`	logical, if TRUE (default) a power of three of the pairwise comparison matrix is used for further calculations.
`zerocor`	logical, if TRUE (default) unobserved combinations (1-0, 0-1) in data for each pair of items are given a frequency of one conf. proposal by Alexandrowicz (2011, p.373).
`ccf`	logical with default `ccf=FALSE` to perform normal item parameter calculation, if set to `ccf=TRUE` just the conditional item (category) frequencies are returned.
`...`	additional parameters passed through.

Details

Parameter calculation is based on the construction of a paired comparison matrix M*nicjc* with entries f*icjc* representing the number of respondents who answered to item *i* in category *c* and to item *j* in category *c-1* widening Choppin's (1968, 1985) conditional pairwise algorithm to polytomous item response formats. This algorithm is simply realized by matrix multiplication.

To avoid numerical problems with off diagonal zero's when constructing the pairwise comparison matrix M*nij*, powers of the M*nicjc* matrix, can be used (Choppin, 1968, 1985). Using powers *k* of M*nicjc* - argument `pot=TRUE` (default), replaces the results of the direct comparisons between *i* and *j* with the sum of the indirect comparisons of *i* and *j* through an intermediate *k*.

In general, it is recommended to use the argument with default value `pot=TRUE`.

Value

A (list) object of class "pair" containing the item category thresholds and difficulties sigma, also called item location.

References

Alexandrowicz, R. W. (2011). 'GANZ RASCH': A Free Software for Categorical Data Analysis. *Social Science Computer Review, 30*(3), 369-379.

Choppin, B. (1968). Item Bank using Samplefree Calibration. *Nature, 219*(5156), 870-872.

Choppin, B. (1985). A fully conditional estimation procedure for Rasch model parameters. *Evaluation in Education, 9*(1), 29-42.

Masters, G. (1982). A rasch model for partial credit scoring. *Psychometrika, 47*(2), 149–174.

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests.* Copenhagen: Danmarks pædagogiske Institut.

Wright, B. D., & Masters, G. N. (1982). *Rating Scale Analysis.* Chicago: MESA Press.

Examples

```
data(bfiN) # loading example data set
# calculating itemparameters for 5 neuroticism items with 6 answer categories (0-5).
neuro_itempar<-pair(daten = bfiN, m = 6)
summary(neuro_itempar)
summary(neuro_itempar, sortdif=TRUE) # ordered by difficulty
# plotting threshold profiles for 5 neuroticism items.
plot(neuro_itempar)
plot(neuro_itempar, sortdif=TRUE) # plotting ordered by difficulty
################ with unequal number of categories
data(sim200x3)
res<-pair(sim200x3)
summary(res)
plot(res)
```

`pairSE` *Item Parameter calculation with Standard Errors for polytomous Partial Credit Model*

Description

Calculation of the item parameter for dichotomous (difficulty) or polytomous items (thurstonian thresholds) and their standard errors (SE) respectively. All parameters are calculatetd using a generalization of the pairwise comparison algorithm (Choppin, 1968, 1985). Missing values up to an high amount in data matrix are allowed, as long as items are proper linked together.

Usage

```
pairSE(daten, m = NULL, nsample = 30, size = 0.5,
  seed = "no", pot = TRUE, zerocor = TRUE,
  verbose = TRUE, ...)
```

Arguments

daten	a data.frame or matrix with optionaly named colums (names of items), potentially with missing values, comprising polytomous or dichotomous (or mixted category numbers) responses of n respondents (rows) on k items (colums) coded starting with 0 for lowest category to *m*-1 for highest category, with *m* beeing a vector (with length k) with the number of categories for the respective item.
m	an integer (will be recycled to a vector of length k) or a vector giving the number of response categories for all items - by default *m* = *NULL* m is calculatetd from data, assuming that every response category is at least once present in data. For sparse data it is strongly recomended to explicitly define the number of categories by defining this argument.
nsample	numeric specifying the number of subsamples sampled from data, which is the number of replications of the parameter calculation. WARNING! specifying high values for nsample (> 100) may result in long computing time without leadig to "better" estimates for SE. This may also be the case when choosing argument size="jack" (see argument size) in combination with large datasets ($N > 5000$).
size	numeric with valid range between 0 and 1 (but not exactly 0 or 1) specifying the size of the subsample of data when bootstraping for SE estimation. As an alternative, size can be set to the character "jack" (size="jack"). This will set the subsample size to *N*-1 and set nsample=N (see argument nsample), with *N* beeing the number of persons in daten.
seed	numeric used for set.seed(seed).
pot	logical, if TRUE (default) a power of three of the pairwise comparison matrix is used for further calculations.
zerocor	logical, if TRUE (default) unobserved combinations (1-0, 0-1) in data for each pair of items are given a frequency of one conf. proposal by Alexandrowicz(2011, p.373).
verbose	logical, if verbose = TRUE (default) a message about subsampling whe calculation standrderrors is sent to console.
...	additional parameters passed through.

Details

Parameter calculation is based on the construction of a paired comparison matrix M*nicjc* with entries f*icjc*, representing the number of respondents who answered to item *i* in category *c* and to item *j* in category *c-1* widening Choppin's (1968, 1985) conditional pairwise algorithm to polytomous item response formats. This algorithm is simply realized by matrix multiplication.

Estimation of standard errors is done by repeated calculation of item parameters for subsamples of the given data.

To avoid numerical problems with off diagonal zeros when constructing the pairwise comparison matrix M*nicjc*, powers of the M*nicjc* matrix, can be used (Choppin, 1968, 1985). Using powers *k* of M*nicjc*, argument pot=TRUE (default), replaces the results of the direct comparisons between *i* and *j* with the sum of the indirect comparisons of *i* and *j* through an intermediate *k*.

In general, it is recommended to use the argument with default value pot=TRUE.

Value

A (list) object of class "pairSE" containing the item category thresholds, difficulties sigma and their standard errors.

A Note on Standard Errors

Estimation of standard errors is done by repeated calculation of item parameters for subsamples of the given data. This procedure is mainly controlled by the arguments `nsample` and `size` (see arguments). With regard to calculation time, the argument `nsample` is the 'time killer'. On the other hand, things (estimation of standard errors) will not necessarily get better when choosing large values for `nsample`. For example choosing `nsample=400` will only result in minimal change for standard error estimation in comparison to (`nsample=30`) which is the default setting (see examples).

References

Choppin, B. (1968). Item Bank using Samplefree Calibration. *Nature, 219*(5156), 870-872.

Choppin, B. (1985). A fully conditional estimation procedure for Rasch model parameters. *Evaluation in Education, 9*(1), 29-42.

Examples

```
data(bfiN) # loading example data set

# calculating itemparameters and their SE for 5 neuroticism items with 6 answer categories (0-5).
neuro_itempar<-pairSE(daten = bfiN, m = 6)
summary(neuro_itempar) # summary for result

# plotting item thresholds with with their CI = 95%
plot(neuro_itempar)
plot(neuro_itempar,sortdif=TRUE)

###### example from details section 'Some Notes on Standard Errors' ########
neuro_itempar_400<-pairSE(daten = bfiN, m = 6,nsample=400)
plot(neuro_itempar)
plot(neuro_itempar_400)
```

`pairwise` *Rasch Model Parameters with pairwise*

Description

The package `pairwise` performs the explicit calculation – not estimation! – of the Rasch item parameters for dichotomous an polytomous response formats using a pairwise comparison approach (Choppin, 1968, 1985).

Details

In case of dichotomous answer formats the item parameter calculation for the Rasch Model (Rasch, 1960), is based on the construction of a pairwise comparison matrix M*nij* with entries f*ij* representing the number of respondents who got item *i* right and item *j* wrong according to Choppin's (1968, 1985) conditional pairwise algorithm.

For the calculation of the item thresholds and difficulty in case of polytomous answer formats, according to the Partial Credit Model (Masters, 1982), a generalization of the pairwise comparison algorithm is used. The construction of the pairwise comparison matrix is therefore extended to the comparison of answer frequencies for each category of each item. In this case, the pairwise

comparison matrix M*nicjc* with entries f*icjc* represents the number of respondents who answered to item *i* in category *c* and to item *j* in category *c-1* widening Choppin's (1968, 1985) conditional pairwise algorithm to polytomous item response formats. Within R this algorithm is simply realized by matrix multiplication.

In general, for both polytomous and dichotomous response formats, the benefit in applying this algorithm lies in it's capability to return stabel item parameter 'estimates' even when using data with a relative high amount of missing values, as long as the items are still proper linked together.

The recent version of the package 'pairwise' computes item parameters for dichotomous and polytomous item responses – and a mixture of both – according the partial credit model using the function `pair`.

Based on the explicit calculated item parameters for a dataset, the person parameters may thereupon be estimated using any estimation approach. The function `pers` implemented in the package uses WLE for estimation of the person parameter.

Author(s)

Joerg-Henrik Heine <jhheine@googlemail.com>

References

Choppin, B. (1968). Item Bank using Samplefree Calibration. *Nature, 219*(5156), 870-872.

Choppin, B. (1985). A fully conditional estimation procedure for Rasch model parameters. *Evaluation in Education, 9*(1), 29-42.

Heine, J. H. & Tarnai, Ch. (2011). Item-Parameter Bestimmung im Rasch-Modell bei unterschiedlichen Datenausfallmechanismen. *Referat im 17. Workshop 'Angewandte Klassifikationsanalyse'* [Item parameter determination in the Rasch model for different missing data mechanisms. Talk at 17. workshop 'Applied classification analysis'], Landhaus Rothenberge, Muenster, Germany 09.-11.11.2011

Heine, J. H., Tarnai, Ch. & Hartmann, F. G. (2011). Eine Methode zur Parameterbestimmung im Rasch-Modell bei fehlenden Werten. *Vortrag auf der 10. Tagung der Fachgruppe Methoden & Evaluation der DGPs.* [A method for parameter estimation in the Rasch model for missing values. Paper presented at the 10th Meeting of the Section Methods & Evaluation of DGPs.] Bamberg, Germany, 21.09.2011 - 23.09. 2011.

Masters, G. N. (1982). A Rasch model for partial credit scoring. *Psychometrika, 47*(2), 149-174.

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests.* Copenhagen: Danmarks pædagogiske Institut.

`pairwise.item.fit` *Item Fit Indicees*

Description

function for calculating item fit indicees

Usage

```
pairwise.item.fit(pers_obj)
```

Arguments

pers_obj	an object of class "pers" as a result from function pers

Details

contrary to many IRT software using Ml based item parameter estimation, pairwise will not exclude persons, showing perfect response vectors (e.g. c(0,0,0) for dataset with three variables), prior to the scaling. Therefor the fit statistics computed with pairwise may deviate somewhat from the fit statistics produced by IRT software using Ml based item parameter estimation (e.g. R-package eRm, depending on the amount of persons with perfect response vectors in the data.

Value

an object of class c("pairwise_item_fit", "data.frame") contaning item fit indicees.

Examples

```
########
data(sim200x3)
result <- pers(pair(sim200x3))
pairwise.item.fit(pers_obj=result) # item fit statistic
```

---

pairwise.person.fit *Person Fit Indicees*

Description

function for calculating person fit indicees

Usage

```
pairwise.person.fit(pers_obj)
```

Arguments

pers_obj	an object of class "pers" as a result from function pers.

Details

contrary to many IRT software using Ml based item parameter estimation, pairwise will not exclude persons, showing perfect response vectors (e.g. c(0,0,0) for dataset with three variables), prior to the scaling. Therefor the fit statistics computed with pairwise may deviate somewhat from the fit statistics produced by IRT software using Ml based item parameter estimation (e.g. R-package eRm, depending on the amount of persons with perfect response vectors in the data.

Value

an object of class c("pairwise_person_fit", "data.frame") contaning person fit indicees.

Examples

```
########
data(sim200x3)
result <- pers(pair(sim200x3))
pairwise.person.fit(pers_obj=result) # item fit statistic
```

---

pers *WLE - Rasch Person Parameter*

Description

This is the (new) main function for calculation of person estimates based on answering dichotomous or polytomous items according theRasch Model (Rasch, 1960) and Partial Credit Model (Masters, 1982), given the item parameters (object of class "pair" - as a result of pair()) and and the datamatrix (argument daten) containing the person respose vectors (rows), using an WL approach, introduced by Warm (1989).

Usage

```
pers(itempar, daten = NULL, incidenz = NULL,
  na_treat = NULL, limit = 1e-05, iter = 50,
  tecout = FALSE)
```

Arguments

itempar	The item parameter prior calculated or estimated. A list object of class "pair" as a result of applying the function pair()) to the data. Or an 'ordinary' "matrix" with nrow = k (number of items) and ncol = m (maximum number of thresholds), holding the 'thurstonian' thresholds of the respective item. some matrix entries may be NA, depending on the number of categories of the respective item.
daten	A "matrix" (or "data.frame") optionaly with named colums (names of items) and named rows (Persone IDs). This argument can be left empty when the argument itempar (above) is of class "pair". daten holds polytomous or dichotomous (or mixted category numbers) responses of n respondents (rows) on k items (colums) coded starting with 0 for lowest category to m-1 for highest category, with m beeing a vector (with length k) with the number of categories for the respective item. Responses in daten must be stored as "integers" (not "factors" !) and may have missing values.
incidenz	This argument is only relevant when items are assigned to different booklets. For such a booklet-design a "matrix" should be assigned to this argument with the same dimensions like daten, containig 0 and 1 integer codes, giving the information (for every person) if the respective item was in the booklet (coded 1) given to the person or not (coded 0).
na_treat	optionaly an integer (vector) defining the type of treatment to missing responses in the argument daten. If set to na_treat=NULL (default) missing responses are treated as missings and the respective person is assigned to an coresponding missing group for estimation. An option is to set na_treat to any integer value between 0 (lowest category) and the numeric code for the maximum ctaegory of the respective item.

limit	numeric giving the limit at which accuracy the WL-algorithm stops.
iter	numeric giving the maximum numer of iteration to perform.
tecout	logical default set toFALSE. If set to TRUE the result will be a (very) long list with estimation details for every case in daten. In case of a booklet-design the list entrys will be divided by "booklet".

Details

no detail in the moment.

Value

An object of class c("pers", "data.frame") or a (very long) "list" (when setting on techout=TRUE) containing the person parameters.

References

Masters, G. (1982). A rasch model for partial credit scoring. *Psychometrika, 47*(2), 149–174.

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests.* Copenhagen: Danmarks pædagogiske Institut.

Warm, T. A. (1989). Weighted likelihood estimation of ability in item response theory. *Psychometrika, 54*(3), 427–450.

Examples

```
############
data(sim200x3)
result <- pers(itempar=pair(sim200x3))
summary(result)
```

---

plot.grm *S3 Plotting Graphical Model Check*

Description

S3 plotting Method for object of class"grm"

Usage

```
  ## S3 method for class 'grm'
plot(x, xymin = NULL, xymax = NULL,
   ci = 2, main = NULL, col.error = "blue",
   itemNames = FALSE, cex.names = 0.8, type = "b",
   xlab = "", ylab = "", pch = 43, las = 3,
   cex.axis = 0.5, ...)
```

Arguments

x	object of class"grm"
xymin	optional lower limit for xy-axis
xymax	optional upper limit for xy-axis
ci	numeric defining confidence intervall for point estimator
main	see plot
col.error	vector of colors for error bars
itemNames	logical wether to plot itemnames
cex.names	magnification factor for itemnames
type	see plot
xlab	see plot
ylab	see plot
pch	see plot
las	see plot
cex.axis	see plot
...	other parameters passed to plot

---

plot.pair *S3 Plotting Thustonian Thresholds*

Description

S3 plotting Method for object of class"pair"

Usage

```
  ## S3 method for class 'pair'
plot(x, sortdif = FALSE, ra = "auto",
   main = NULL, col.lines = (1:dim(x$threshold)[2]),
   type = "b", xlab = "items", ylab = "logits",
   pch = (1:dim(x$threshold)[2]), las = 3, cex.axis = 0.8,
   ...)
```

Arguments

x	object of class"pair"
sortdif	logical wether to order items by difficulty
ra	either the character "auto" (default) or an numeric, defining the (logit) range for y-axis
main	see plot
col.lines	vector of colors for threshold profile lines
type	see plot
xlab	see plot
ylab	see plot

pch	see plot
las	see plot
cex.axis	see plot
...	other parameters passed to plot

---

plot.pairSE *S3 Plotting Thustonian Thresholds with SE*

Description

S3 plotting Method for object of class"pairSE"

Usage

```
  ## S3 method for class 'pairSE'
plot(x, ci = 2, sortdif = FALSE,
   ra = "auto", main = NULL,
   col.lines = 1:(dim(x$parameter)[2] - 1),
   col.error = 1:(dim(x$parameter)[2] - 1), type = "b",
   xlab = "items", ylab = "logits", pch = 20, las = 3,
   cex.axis = 0.8, ...)
```

Arguments

x	object of class"pairSE"
sortdif	logical wether to order items by difficulty
ra	either the character "auto" (default) or an numeric, defining the (logit) range for y-axis
ci	numeric defining confidence intervall for point estimator
main	see plot
col.lines	vector of colors for threshold profile lines
col.error	vector of colors for error bars
type	see plot
xlab	see plot
ylab	see plot
pch	see plot
las	see plot
cex.axis	see plot
...	other parameters passed to plot

ptbis *Point Biserial Correlations*

Description

Calculation of the point biserial correlations for dicho- or polytomous item categories with total scale (person parameter).

Usage

```
ptbis(y, daten = NULL)
```

Arguments

y	either an object of class "pers", or an numeric vector as an result of any scaling approach (WLE, MLE, Rawscore, etc.) relating to the Items (columns) in daten.
daten	if argument y is not an object of class "pers", a data.frame, potentially with missing values, comprising dicho- or polytomous items (columns).

Details

to come ...

Value

An object of class "data.frame" and "ptbis" containing item statistics.

Examples

```
######################
########
data(sim200x3) # loading reponse data
y <- rowSums(sim200x3)
ptbis(y=y, daten=sim200x3)
####
result <- pers(pair(sim200x3))
ptbis(y= result)
```

sim200x3 *Simulated Data*

Description

Simulated data for 200 'subjects' 'answering' to 3 items with unequal number of categories – one dichotomous and two polytoumous items.

Usage

```
data(sim200x3)
```

Format

A data.frame containing 3 variables and 200 obsevations.

Details

This simulated data is used as an example in the rasch module of the 'ALMO - Statistiksystem'.

Source

http://www.almo-statistik.de/

References

To come ...

Examples

```
data(sim200x3)
dim(sim200x3)
############################################################
apply(sim200x3,2,table)
```

---

summary.pers *S3 Summary for Thetas*

Description

S3 summary Method for object of class"pers"

Usage

```
## S3 method for class 'pers'
summary(object, sortwle = FALSE, ...)
```

Arguments

object	object of class"pers"
sortwle	logical wether to order persons by ability
...	other parameters passed trough

Index

∗Topic **datasets**
bfiN, 2
bfiN_miss, 3
cog, 5
cogBOOKLET, 5
DEU_PISA2012, 6
sim200x3, 23

bfiN, 2
bfiN_miss, 3

catprob, 4
cog, *5*, 5
cogBOOKLET, 5, *12*

DEU_PISA2012, 6

esc, 7

ftab, 8

gif, 8
grm, 9

make.incidenz, 11

pair, *4*, 13, *17*, *19*
pairSE, *9–11*, 14
pairwise, 16
pairwise-package *(*pairwise*)*, 16
pairwise.item.fit, 17
pairwise.person.fit, 18
pers, *7*, *9*, *12*, *17, 18*, 19
plot, *7*, *21, 22*
plot.grm, 20
plot.pair, 21
plot.pairSE, 22
ptbis, 23

sim200x3, 23
summary.pers, 24