Package 'plink'

July 23, 2025

Version 1.5-1

Date 2017-04-26

Title IRT Separate Calibration Linking Methods

Depends R (>= 3.3.3), methods, lattice

Imports MASS, statmod

Author Jonathan P. Weeks <weeksjp@gmail.com>

Maintainer Jonathan P. Weeks <weeksjp@gmail.com>

Description Item response theory based methods are used to compute
linking constants and conduct chain linking of unidimensional
or multidimensional tests for multiple groups under a common
item design. The unidimensional methods include the Mean/Mean,
Mean/Sigma, Haebara, and Stocking-Lord methods for dichotomous
(1PL, 2PL and 3PL) and/or polytomous (graded response, partial
credit/generalized partial credit, nominal, and multiple-choice
model) items. The multidimensional methods include the least
squares method and extensions of the Haebara and Stocking-Lord
method using single or multiple dilation parameters for
multidimensional extensions of all the unidimensional
dichotomous and polytomous item response models. The package
also includes functions for importing item and/or ability
parameters from common IRT software, conducting IRT true score
and observed score equating, and plotting item response
curves/surfaces, vector plots, information plots, and comparison
plots for examining parameter drift.

LazyLoad Yes

LazyData Yes

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2017-04-26 16:36:59 UTC

Contents

---

plink-package   *IRT Separate Calibration Linking Methods*

Description

This package uses unidimensional and multidimensional item response theory methods to compute linking constants and conduct chain linking of tests for multiple groups under a nonequivalent groups common item design.

Details

The package consists of three types of functions:

- Response Probability Functions: These functions compute response probabilities for specified theta values (for as many dimensions and theta values as allowed by your computer's memory). The package can estimate probabilities for the item response models listed below. For all of these models, I have tried to include examples from the corresponding journal articles as a way to show that the equations are correct.
  - 1PL
  - 2PL
  - 3PL
  - Graded Response Model* (cumulative or category probabilities)
  - Partial Credit Model*
  - Generalized Partial Credit Model*
  - Nominal Response Model
  - Multiple-Choice Model
  - M1PL
  - M2PL
  - M3PL
  - MD Graded Response Model* (cumulative or category probabilities)
  - MD Partial Credit Model*
  - MD Generalized Partial Credit Model*
  - MD Nominal Response Model
  - MD Multiple-Choice Model (this model has not formally been presented in the literature)

  * These models can be specified using a location parameter

- Linking Function: There is only one linking function, but it includes a variety of linking methods. The most notable feature of this function (as compared to other software) is that it allows you to input parameters for more than two groups and then chain link all of the tests together in a single run. Below are several options.
  - Symmetric or Non-Symmetric linking (as originally presented by Haebara)
  - Specification of weights (can include uniform, quadrature, and normal density weights for default or specified theta points)
  - Choice of base group
  - Choice of method to rescale item parameters and/or ability estimates (if included)

  When the item parameters correspond to a unidimensional model, linking constants can be estimated using the following methods
  - Mean/Mean
  - Mean/Sigma
  - Haebara
  - Stocking-Lord

  When the item parameters correspond to a multidimensional model, linking constants can be estimated using the following methods (with different options for the first two)
  - Multidimensional extension of Haebara

- – Multidimensional extension of Stocking-Lord
- – Reckase-Martineau (based on the oblique procrustes rotation and least squares method for estimating the translation vector m.

For the first two approaches there is a "dilation" argument where

- – all of the elements of the rotation/scaling matrix A and the translation vector m are estimated via the optimization routine (see Oshima, Davey, & Lee, 2000)
- – an orthongonal procrustes rotation to resolve the rotational indeterminacy and a single dilation parameter is estimated (see Li & Lissitz, 2000)
- – an orthongonal procrustes rotation to resolve the rotational indeterminacy and different parameters are estimated to adjust the scale of each dimension (see Min, 2003).

For all of these approaches an optional matrix can be specified to identify the ordering of dimensions. It allows you to specify an overall factor structure that can take into account construct shift or different orderings of the factor loadings.

- • Utility Functions:
  - – Import item parameters and/or ability estimates from BILOG-MG 3, PARSCALE 4, MULTILOG 7, TESTFACT 4, ICL, BMIRT, and the eRm and ltm packages.
  - – Separate item parameters from matrices or lists into the slope, difficulty, category, etc. values for use in computing response probabilities, descriptive statistics, etc.
  - – Combine multiple sets of parameters into a single object for use in the linking function (this essentially creates a blueprint of all the items including the IRT model used for each item, the number of response categories, and the mapping of common items between groups)
  - – Plot item characteristic/category curves (for unidimensional items), item response/category surfaces, contour plots, level plots, and three types of vector plots (for multidimensional items). The first three types of multidimensional plots include functionality for creating "conditional" surfaces when there are more than two dimensions).
  - – Summarize the item parameters (unique and/or common) for each item response model separately, and overall

Running the separate calibration is typically a two-step process. The first step is to format the item parameters and the second step is to run the function plink. In the simplest scenario, the parameters should be formatted as a single irt.pars object with multiple groups. Refer to the function as.irt.pars for specific details. Once in this format, the linking constants can be computed using plink. The summary function can be used to summarize the common item parameters (including descriptive statistics) and the linking constants.

To compute response probabilities for a given model, the following functions can be used: drm, gpcm, grm, mcm, or nrm. The plot function can be used to create item/category characteristic curves, item/category response surfaces, and vector plots.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL http://www.jstatsoft.org/v35/i12/

---

act.mcm   *ACT Mathematics Test 1997-1998 Multiple-Choice Model*

Description

This (unidimensional) dataset includes multiple-choice model item parameter estimates for two groups. There are 60 items, all of which are common items. The estimates are from the 1997 and 1998 ACT mathematics test. The item parameters were obtained from the examples for the software mceq written by Bradley Hanson.

Usage

```
act.mcm
```

Format

A list of length two. The matrices in the two list elements contain multiple-choice model item parameters for 1997 and 1998 respectively. The first six columns in each matrix are the slope parameters, the next six columns are the category difficulty parameters, and the last five columns are the lower asymptote parameters.

Source

Kim, J.-S. & Hanson, B. A. (2002). Test Equating Under the Multiple-Choice Model. *Applied Psychological Measurement*, *26*(3), 255-270.

The item parameters can be found here http://www.b-a-h.com/software/mcmequate/index.html

---

act.nrm   *ACT Mathematics Test 1997-1998 Nominal Response Model*

Description

This (unidimensional)dataset includes nominal response model item parameter estimates for two groups. There are 60 items, all of which are common items. The estimates are from the 1997 and 1998 ACT mathematics test. The item parameters were obtained from the examples for the software nreq written by Bradley Hanson.

Usage

```
act.nrm
```

Format

A list of length two. The matrices in the two list elements contain nominal response model item parameters for 1997 and 1998 respectively. The first six columns in each matrix are the slope parameters and the next six columns are the category difficulty parameters.

Source

Kim, J.-S. & Hanson, B. A. (2002). Test Equating Under the Multiple-Choice Model. *Applied Psychological Measurement*, *26*(3), 255-270.

The item parameters can be found here http://www.b-a-h.com/software/mcmequate/index.html

---

as.irt.pars   *irt.pars objects*

Description

This function attempts to turn the given values into an `irt.pars` object that is used primarily with the `plink` function.

Usage

```
as.irt.pars(x, common, cat, poly.mod, dimensions = 1,
  location = FALSE, grp.names, ...)

## S4 method for signature 'numeric', 'missing'
as.irt.pars(x, common, cat, poly.mod, dimensions, location,
  grp.names, ...)

## S4 method for signature 'matrix', 'missing'
as.irt.pars(x, common, cat, poly.mod, dimensions, location,
  grp.names, ...)

## S4 method for signature 'data.frame', 'missing'
as.irt.pars(x, common, cat, poly.mod, dimensions, location,
  grp.names, ...)

## S4 method for signature 'list', 'missing'
as.irt.pars(x, common, cat, poly.mod, dimensions, location,
  grp.names, ...)

## S4 method for signature 'sep.pars', 'missing'
as.irt.pars(x, common, cat, poly.mod, dimensions, location,
  grp.names, ...)

## S4 method for signature 'list', 'matrix'
as.irt.pars(x, common, cat, poly.mod, dimensions, location,
  grp.names, ...)

## S4 method for signature 'list', 'list'
as.irt.pars(x, common, cat, poly.mod, dimensions, location,
  grp.names, ...)
```

Arguments

x	an R object containing item parameters. For details on the formatting of parameters for specific item response models see the corresponding methods (i.e., [drm](), [gpcm](), [grm](), [mcm](), and [nrm]()). See below for details on combining parameters from multiple models.
common	j x 2 matrix or list of matrices identifying the common items between adjacent groups in x. This argument is only applicable when x includes two or more groups
cat	vector or list of the number of response categories for each item in x. if x is a list, cat should be a list as well. For multiple-choice model items, cat is the number of response categories plus one (the additional category is for 'do not know')
poly.mod	a [poly.mod]() object or a list of poly.mod objects (one for each group in x)
dimensions	numeric vector identifying the number of modeled dimensions in each group.
location	logical vector identifying whether the parameters in x include a location parameter (i.e., for polytomous items)
grp.names	character vector of group names
...	further arguments passed to or from other methods

Value

Returns an object of class [irt.pars]()

Methods

x = "numeric", common = "missing" This method should only be used for the Rasch model where x is a vector of difficulty parameters (or parameters related to item difficulty in the multidimensional case). Under this method the slopes and lower asymptote values for all items will default to one and zero respectively. This is true for both the unidimensional and multidimensional case.

x = "matrix", common = "missing" x can include item parameters from multiple models. The general format for structuring x is an additive column approach. That is, the left-most columns are typically for discrimination/slope parameters, the next column, if applicable, is for location parameters, the next set of columns is for difficulty/threshold/step/category parameters, and the final set of columns is for lower asymptote (guessing) parameters. When multiple models are included, or models with differing numbers of response categories, not all cells in x may have data. In these instances, cells with no data should be NA. The resulting matrix will be an n x k matrix for n items and k equal to the maximum number of columns (across all item response models). In essence, the combination of multiple models is equivalent to formatting the item parameters for each response model separately, stacking the matrices on top of one another and then filling in any missing cells with NAs.

x = "data.frame", common = "missing" See the method for x = "matrix"

x = "list", common = "missing" This method can include a list with one, two, or three elements. In general, these elements correspond to discrimination/slope parameters, difficulty/threshold/ step/category parameters, and lower asymptote (guessing) parameters, although this may not

be the case depending on the supplied parameters. If a combination of models are used, the number of list elements should correspond with the maximum number of elements across models. For example, if the 3PL model (3 list elements) and nominal response model (2 list elements) are used, the list should include three elements. Within each list element, the parameters should be formatted as vectors or matrices (depending on the specific item response models). For the format of the matrices, see the method for x = "matrix" above and/or the methods corresponding to the specific response models. When location parameters are included, these values should be placed in the first column of the matrix of difficulty/threshold/step/category parameters (in the corresponding list element).

x = "sep.pars", common = "missing" x is an object of class `sep.pars`. The arguments `cat`, `poly.mod`, and `location` do not need to be included.

x = "list", common = "matrix" This method is intended for the creation of an object with only two groups. Each list element should conform to one of the formats listed above (i.e. a numeric vector, matrix/data.frame, list, or sep.pars object) or be an `irt.pars` object. The format of each list element does not need to be the same. That is, the first list element might be a `"sep.pars"` object while the second element is a list of item parameters. If an object of class `irt.pars` is included, the object can contain information for a single group or for multiple groups. If the `irt.pars` object includes multiple groups, `common` should identify the common items between the last group in `x[[1]]` and the first group in `x[[2]]`.

For this method `common` is a j x 2 matrix where the first column identifies the common items for the first group (`x[[1]]`). The second column in `common` identifies the corresponding set of common items from the next list element. For example, if item 4 in group one (row 4 in `x[[1]]`) is the same as item 6 in group two, the first row of `common` would be `"4,6"`.

If the objects in x are of class sep.pars or irt.pars, `cat`, `poly.mod`, and `location` do not need to be included.

x = "list", common = "list" This method is intended for the creation of an object with more than two groups; however, if there are two groups `common` can be defined as a list with length one. Each list element should conform to one of the formats listed above (i.e. a numeric vector, matrix/data.frame, list, or sep.pars object) or be an `irt.pars` object. The format of each list element does not need to be the same. That is, the first list element might be a `"sep.pars"` object while the second element is a list of item parameters. If an object of class `irt.pars` is included, the object can contain information for a single group or for multiple groups. The list elements in x should be ordered such that adjacent elements correspond to adjacent groups. If an `"irt.pars"` object is included with multiple groups, the list element following this object should contain information for a group that is adjacent to the last group in the `"irt.pars"` object

For this method `common` is a list of j x 2 matrices where the first column identifies the identifies the common items for the first group of two adjacent list elements in The second column in `common` identifies the corresponding set of common items from the next list element in x. For example, if x contains only two list elements, a single set of common items links them together. If item 4 in group one (row 4 in slot `pars`) is the same as item 6 in group two, the first row of `common` would be `"4,6"`.

If all the objects in x are of class sep.pars or irt.pars, the arguments `cat`, `poly.mod`, and `location` do not need to be included.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL http://www.jstatsoft.org/v35/i12/

See Also

irt.pars, as.poly.mod, poly.mod, sep.pars

Examples

```
# Create object for three dichotomous (1PL) items with difficulties
# equal to -1, 0, 1
x <- as.irt.pars(c(-1,0,1))


# Create object for three dichotomous (3PL) items and two polytomous
# (gpcm) items without a location parameter
# (use signature matrix, missing)
dichot <- matrix(c(1.2, .8, .9, 2.3, -1.1, -.2, .24, .19, .13),3,3)
poly <- matrix(c(.64, -1.8, -.73, .45, NA, .88, .06, 1.4, 1.9, 2.6),
  2,5,byrow=TRUE)
pars <- rbind(cbind(dichot,matrix(NA,3,2)),poly)
cat <- c(2,2,2,4,5)
pm <- as.poly.mod(5, c("drm","gpcm"), list(1:3,4:5))
x <- as.irt.pars(pars, cat=cat, poly.mod=pm)
summary(x)


# Create object for three dichotomous (3PL) items and two polytomous
# (gpcm) items without a location parameter
# (use signature list, missing)
a <- c(1.2, .8, .9, .64, .88)
b <- matrix(c(
  2.3, rep(NA,3),
  -1.1, rep(NA,3),
  -.2, rep(NA,3),
  -1.8, -.73, .45, NA,
  .06, 1.4, 1.9, 2.6),5,4,byrow=TRUE)
c <- c(1.4, 1.9, 2.6, NA, NA)
pars <- list(a,b,c)
cat <- c(2,2,2,4,5)
pm <- as.poly.mod(5, c("drm","gpcm"), list(1:3,4:5))
x <- as.irt.pars(pars, cat=cat, poly.mod=pm)
summary(x)


# Create object for three dichotomous (3PL) items, four polytomous items,
# two gpcm items and two nrm items. Include a location parameter for the
# gpcm items (use signature list, missing)
a <- matrix(c(
  1.2, rep(NA,4),
  .8, rep(NA,4),
```

```
    .9, rep(NA,4),
    .64, rep(NA,4),
    .88, rep(NA,4),
    .905, .522, -.469, -.959, NA,
    .828, .375, -.357, -.079, -.817),7,5,byrow=TRUE)
b <- matrix(c(
  2.3, rep(NA,4),
  -1.1, rep(NA,4),
  -.2, rep(NA,4),
  -.69, -1.11, -.04, 1.14, NA,
  1.49, -1.43, -.09, .41, 1.11,
  .126, -.206, -.257, .336, NA,
  .565, .865, -1.186, -1.199, .993),7,5,byrow=TRUE)
c <- c(.14, .19, .26, rep(NA,4))
pars <- list(a,b,c)
cat <- c(2,2,2,4,5,4,5)
pm <- as.poly.mod(7, c("drm","gpcm","nrm"), list(1:3,4:5,6:7))
x <- as.irt.pars(pars, cat=cat, poly.mod=pm, location=TRUE)
summary(x, TRUE)


# Create object with two groups (all dichotomous items)
pm <- as.poly.mod(36)
x <- as.irt.pars(KB04$pars, KB04$common, cat=list(rep(2,36),rep(2,36)),
  list(pm,pm), grp.names=c("form.x","form.y"))
summary(x, TRUE)


# Create object with six groups (all dichotomous items)
pars <- TK07$pars
common <- TK07$common
cat <- list(rep(2,26),rep(2,34),rep(2,37),rep(2,40),rep(2,41),rep(2,43))
pm1 <- as.poly.mod(26)
pm2 <- as.poly.mod(34)
pm3 <- as.poly.mod(37)
pm4 <- as.poly.mod(40)
pm5 <- as.poly.mod(41)
pm6 <- as.poly.mod(43)
pm <- list(pm1, pm2, pm3, pm4, pm5, pm6)
x <- as.irt.pars(pars, common, cat, pm,
  grp.names=paste("grade",3:8,sep=""))

# Create an object with two groups using mixed-format items and
# a mixed placement of common items. This example uses the dgn dataset.
pm1=as.poly.mod(55,c("drm","gpcm","nrm"),dgn$items$group1)
pm2=as.poly.mod(55,c("drm","gpcm","nrm"),dgn$items$group2)
x=as.irt.pars(dgn$pars,dgn$common,dgn$cat,list(pm1,pm2))
summary(x, TRUE)
```

---

as.poly.mod   *poly.mod objects*

Description

This function attempts to turn the given values into a poly.mod object that associates each item with a specific unidimensional or multidimensional item response model.

Usage

```
as.poly.mod(n, model = "drm", items = NULL)
```

Arguments

n	total number of items
model	character vector identifying the IRT models used to estimate the item parameters. The only acceptable models are drm, gpcm, grm, mcm, and nrm. See below for an explanation of the codes.
items	list identifying the item numbers from a set of parameters that correspond to the given model in model.

Details

When creating a poly.mod object, there is no difference in the specification for unidimensional versus multidimensional item response models. If all the items are dichotomous, it is only necessary to specify a value for n. If all the items correspond to a single model (other than drm), only n and model need to be specified.

The IRT models associated with the codes:

drm: dichotomous response models (includes the 1PL, 2PL, 3PL, M1PL, M2PL, and M3PL)

gpcm: partial credit model, generalized partial credit model, multidimensional partial credit model, and multidimensional generalized partial credit model

grm: graded response model and multidimensional graded response model

mcm: multiple-choice model and multidimensional multiple-choice model

nrm: nominal response model and multidimensional nominal response model

Value

Returns an object of class poly.mod

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL <http://www.jstatsoft.org/v35/i12/>

See Also

poly.mod

Examples

```
# Ten dichotomous items
as.poly.mod(10)

# The first ten items in the set of associated (not present here) item
# parameters are dichotomous and the last five were estimated using the
# generalized partial credit model
as.poly.mod(15, c("drm", "gpcm"), list(1:10,11:15) )

# Ten multidimensional graded response model items
# Note: This same specification would be used for a unidimensional
# graded response model
as.poly.mod(10, "grm")
```

---

as.weight   *Calibration Weights*

Description

This function facilitates the creation of either a set of quadrature weights or weights based on a set of theta values for use in the function [plink](#)

Usage

```
as.weight(n, theta, weight, quadrature = FALSE, normal.wt = FALSE,
  dimensions = 1, ...)
```

Arguments

n	numeric value or vector identifying the number of theta values to use for each dimension. If only one value is supplied but dimensions is greater than one, n will be the same for all dimensions.
theta	vector or list of theta values. If dimensions is greater than one, list elements should correspond to each dimension, otherwise the single vector will be repeated for all dimensions.
weight	vector or list of weights. If dimensions is greater than one, list elements should correspond to each dimension, otherwise the single vector will be repeated for all dimensions.
quadrature	if TRUE, quadrature points and weights will be computed for the corresponding number of points n for each dimension respectively. See [gauss.quad.prob](#) for more information on creating quadrature points and weights.
normal.wt	if TRUE and weight is missing, the weights for theta will be computed to correspond to the densities from a normal distribution.
dimensions	number of dimensions for which the weights should be created
...	further arguments passed to other methods

Details

When weighting expected response probabilities at different theta values using characteristic curve linking methods, there are a variety of approaches one can take. These range from uniform weights to normal weights, to quadrature weights corresponding to some a priori specified distribution. The purpose of this function is to facilitate the creation of these weights for use in `plink` .

For all approaches, when more than one dimension is specified, the weights for each combined set of theta values will be a multiplicative weight. For example, if there are two dimensions and the specified weights corresponding to two specific theta values on each dimension respectively are 0.8 and 1.2, the final weight for this pair of theta values will be 0.96.

Uniform Weights Five general approaches can be used to create uniform weights.

- If no arguments are supplied, a set of weights (all equal to one) will be returned for a single dimension, for 40 equal interval theta values ranging from -4 to 4. If dimensions is greater than one, seven equal interval theta values ranging from -4 to 4 will be specified for each dimension. For instance, for two dimensions, there will be weights for 7 x 7 (49) points.
- If only a value for n is supplied, uniform weights (all equal to one) will be created for n points ranging from -4 to 4 for each dimension specified.
- If values are only specified for `theta`, uniform weights (all equal to one) will be created for each of these values for each dimension specified.
- If values are only specified for `weight` where the values are all equal. In this case, equal interval theta values will be selected from -4 to 4 to correspond to the number of specified weights.
- If values are specified for `theta` and uniform values are specified for `weight`.

Non-Uniform Weights Four general approaches can be used to create non-uniform weights.

- If values are only specified for `weight` where the values are not equal. In this case, equal interval theta values will be selected from -4 to 4 to correspond to the number of specified weights.
- If values are specified for `theta` and varying values are specified for `weight`.
- If quadrature is equal to TRUE, and no other arguments are specified n will default to 40. if dimensions is greater than one, n will default to seven for each dimension. In either case n quadrature points and weights will be selected from a standard normal distribution. To change the mean and/or standard deviation of this distribution, values for mu and sigma respectively should be specified. See `gauss.quad.prob` for more information. Different means and/or SDs can be supplied for each dimension. If values are specified for `theta` or `weight`, the quadrature approach will not be used.
- If quadrature equals TRUE other distributions can be specified for n points. See `gauss.quad.prob` for more information.
- If normal.wt equals TRUE, normal densities will be created for the specified `theta` values (if supplied) or equal interval values. The default distribution will be standard normal, but different means and/or standard deviations can be specified by passing arguments for mean and sd respectively. Different means and/or SDs can be supplied for each dimension. If no values are included for `theta`, equal interval theta values will be created for the range of three SDs above and below the mean. If values are specified for `weight`, the `normal.wt` argument will be ignored.

See Kolen & Brennan (2004) for more information on calibration weights.

Value

Returns a list of length two. The first list element is an n x m matrix of theta values for m dimensions. The second list element is a vector of weights corresponding to the theta values in the first list element.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Kolen, M. J., & Brennan, R. L. (2004) *Test Equating, Scaling, and Linking.* New York: Springer

See Also

plink, gauss.quad.prob

Examples

```
###### Unidimensional Examples ######
# Create a set of 40 (default) theta values and uniform weights
wt <- as.weight()

# Create a set of 40 quadrature points and weights using a standard
# normal distribution
wt <- as.weight(quadrature=TRUE)

# Create a set of 30 quadrature points and weights using a normal
# distribution with a mean of 0.5
wt <- as.weight(n=30, quadrature=TRUE, mu=0.5)

# Create weights for a set of random normal theta values
wt <- as.weight(theta=rnorm(100))

# Create an object with equal interval theta values and standard
# normal density weights
wt <- as.weight(theta=seq(-4,4,0.05), normal.wt=TRUE)

# Create an object with equal interval theta values and normal
# density weights with a mean of 0.5 and SD equal to .92
wt <- as.weight(theta=seq(-4,4,0.05), normal.wt=TRUE, mean=0.5, sd=0.92)

###### Multidimensional Examples ######
# Create a set of 49 theta values and uniform weights
# (based on seven points for each dimension)
wt <- as.weight(dimensions=2)

# Create a set of 100 quadrature points and weights using a normal
# distribution with a means of 0 and 0.5 for the two dimensions respectively
wt <- as.weight(n=10, quadrature=TRUE, mu=c(0,0.5), dimensions=2)

# Create an object with equal interval theta values and standard
```

```
# normal density weights for three dimensions
wt <- as.weight(theta=seq(-3,3), normal.wt=TRUE, dimensions=3)

# Create an object with two sets of equal interval theta values for
# two dimensions
wt <- as.weight(theta=list(seq(-4,4),seq(-3,3)), dimensions=2)

# Create an object with two sets of random normal theta values and
# standard normal density weights for two dimensions
wt <- as.weight(theta=list(rnorm(10),rnorm(10)), normal.wt=TRUE, dimensions=2)
```

---

combine.pars   *Combine Item Parameters for Multiple Groups*

Description

This function combines objects of class `irt.pars` and/or `sep.pars` into a single irt.pars object

Usage

```
combine.pars(x, common, grp.names)
```

Arguments

x	an ordered list containing two or more `irt.pars` objects, two or more `sep.pars` objects, or a combination of `irt.pars` and `sep.pars` objects.
common	an n x 2 matrix or list of matrices identifying the common items between adjacent pairs of objects in x. See below for more details.
grp.names	character vector of names for all the groups in the returned object

Details

Although many of the methods in this package allow for lists containing `irt.pars` and `sep.pars` objects, it may be helpful to combine the item parameters for multiple groups into a single object. x can include a combination of `irt.pars` and `sep.pars` objects. The `irt.pars` objects can contain information for a single group or multiple groups. The list elements in x should be ordered such that adjacent elements correspond to adjacent groups. If an `irt.pars` object is included with multiple groups, the list element following this object should contain information for a group that is adjacent to the last group in the `irt.pars` object.

If x contains only two elements, common should be a matrix. If x contains more than two elements, common should be a list. In any of the common matrices the first column identifies the common items for the first group of two adjacent list elements in x. The second column in common identifies the corresponding set of common items from the next list element in x. For example, if x contains only two list elements, a single set of common items links them together. If item 4 in group one (row 4 in slot pars) is the same as item 6 in group two, the first row of common would be "4,6".

If an `irt.pars` object is included with multiple groups, common should correspond to the set of common items between the last group in the `irt.pars` object and the group in the adjacent element in x.

Value

Returns an object of class `irt.pars`

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL http://www.jstatsoft.org/v35/i12/

See Also

`irt.pars`, `as.irt.pars`, `sep.pars`, `sep.pars`

---

dgn   *Unidimensional Mixed-Format Item Parameters*

Description

This (unidimensional) dataset includes item parameters for two groups estimated under the three parameter logistic model, the generalized partial credit model, and the nominal response model. There are 15 common items (7 drm, 3 gpcm, 5 nrm). This dataset is for illustrative purposes to show how items from different response models can be mixed together and common items can be in different positions across groups.

Usage

dgn

Format

A list of length four. The first element is a list of length two with item parameter estimates for groups 1 and 2 respectively. The gpcm items do not include a location parameter. The second list element is a list identifying the number of response categories for the two groups. The third element, specifies which items correspond to the different item response models for each group respectively. The last element is a matrix of common items between the two groups.

References

The item parameters are loosely based on parameters listed in the following articles

Kim, J.-S. (2006). Using the Distractor Categories of Multiple-Choice Items to Improve IRT Linking. *Journal of Educational Measurement*, *43*(3), 193-213.

Kim, S. & Lee, W.-C. (2006). An Extension of Four IRT Linking Methods for Mixed-Format Tests. *Journal of Educational Measurement*, *43*(1), 53-76.

---

drm-methods   *Dichotomous Response Model Probabilities*

Description

This function computes the probability of correct responses (and optionally, incorrect responses) for one or more items for a given set of theta values using the 1PL, 2PL, 3PL, M1PL, M2PL or M3PL model, depending on the included item parameters and the specified number of dimensions.

Usage

```
drm(x, theta, dimensions = 1, D = 1, incorrect = FALSE,
  print.mod = FALSE, items, information = FALSE, angle, ...)

## S4 method for signature 'numeric'
drm(x, theta, dimensions, D, incorrect, print.mod, items, information, angle, ...)

## S4 method for signature 'matrix'
drm(x, theta, dimensions, D, incorrect, print.mod, items, information, angle, ...)

## S4 method for signature 'data.frame'
drm(x, theta, dimensions, D, incorrect, print.mod, items, information, angle, ...)

## S4 method for signature 'list'
drm(x, theta, dimensions, D, incorrect, print.mod, items, information, angle, ...)

## S4 method for signature 'irt.pars'
drm(x, theta, dimensions, D, incorrect, print.mod, items, information, angle, ...)

## S4 method for signature 'sep.pars'
drm(x, theta, dimensions, D, incorrect, print.mod, items, information, angle, ...)
```

Arguments

x	an R object containing item parameters
theta	vector, matrix, or list of theta values for which probabilities will be computed. If theta is not specified, an equal interval range of values from -4 to 4 is used with an increment of 0.5. See details below for more information.
dimensions	number of modeled dimensions
D	scaling constant. The default value assumes that the parameters are already in the desired metric. If the parameters are in the logistic metric, they can be transformed to a normal metric by setting $D = 1.7$
incorrect	if TRUE, compute probabilities for the incorrect response category)
print.mod	if TRUE, print the model that was used (i.e., 1PL, 2PL, 3PL, M1PL, M2PL, or M3PL)

items	numeric vector identifying the items for which probabilities should be computed
information	logical value. If TRUE compute item information. In the multidimensional case, information will be computed in the directions specified by angle or default angles of 0 - 90 in increments of 10 degrees.
angle	vector or matrix of angles between the dimension 1 axis and the corresponding axes for each of the other dimensions for which information will be computed. When there are more than two dimensions and angle is a vector, the same set of angles will be used relative to each of the corresponding axes.
...	further arguments passed to or from other methods

Details

theta can be specified as a vector, matrix, or list. For the unidimensional case, theta should be a vector. If a matrix or list of values is supplied, they will be converted to a single vector of theta values. For the multidimensional case, if a vector of values is supplied it will be assumed that this same set of values should be used for each dimension. Probabilities will be computed for each combination of theta values. Similarly, if a list is supplied, probabilities will be computed for each combination of theta values. In instances where probabilities are desired for specific combinations of theta values, a j x m matrix should be specified for j ability points and m dimensions where the columns are ordered from dimension 1 to m.

Value

Returns an object of class `irt.prob`

Methods

x = "numeric" This method should only be used for the Rasch model where x is a vector of difficulty parameters (or parameters related to item difficulty in the multidimensional case). Under this method the slopes and lower asymptote values for all items will default to one and zero respectively. This is true for both the unidimensional and multidimensional case.

x = "matrix" This method allows one to specify an n x k matrix for n items, for m dimensions, and k equal to m, m+1, or m+2. The first m columns generally correspond to the discrimination/slope parameters, and the last two columns correspond to the difficulty, and lower asymptote parameters, although this may not be the case depending on the supplied parameters.

1PL/M1PL: For the 1PL and M1PL model with discriminations equal to 1 (Rasch Model), an n x 1 matrix of item difficulties can be supplied. An n x (m+1) matrix can also be used with all the values in the first m columns equal to 1 and difficulty parameters in the last column. For discrimination values other than 1, x should include at least m+1 columns where the first m columns contain the item discriminations (identical for all items) and the last column is for the item difficulties. The lower asymptote defaults to zero for all items; however, m+2 columns can be specified where the values in the last column all equal zero.

2PL/M2PL: x should include at least m+1 columns where the first m columns contain the item discriminations and the last column is for the item difficulties. The lower asymptote defaults to zero for all items; however, m+2 columns can be specified where the values in the last column all equal zero.

3PL/M3PL: x should include m+2 columns where the first m columns contain the item discriminations, the m+1 column is for item difficulties, and the last column is for lower asymptote values.

x = "data.frame" See the method for x = "matrix"

x = "list" This method can include a list with 1, 2, or 3 elements. In general, these elements correspond to discrimination, difficulty, and lower asymptote parameters, although this may not be the case depending on the supplied parameters. For the unidimensional case, each list element can be a vector of length n or an n x 1 matrix for n items. For the multidimensional case, the list element for the discrimination parameters should be an n x m matrix for m dimensions. The other list elements can be vectors of length n or an n x 1 matrix for n items.

1PL/M1PL: For the 1PL and M1PL model with discriminations equal to 1 (Rasch Model), one element with item difficulties can be supplied. Alternatively, two elements can be used with the first list element containing a matrix/vector of ones and difficulty parameters in the second list element. For discrimination values other than 1, x should contain at least two list elements where the first contains the item discriminations (identical for all items) and the second is for item difficulties. The lower asymptote defaults to zero for all items; however, a third element with a vector/matrix of zeros can be included.

2PL/M2PL: x should contain at least two list elements where the first element contains the item discriminations and the second element includes the item difficulties. The lower asymptote defaults to zero for all items; however, a third element with a vector/matrix of zeros can be included.

3PL/M3PL: x should include three list elements where the first element contains the item discriminations, the second element includes the item difficulties, and the third element contains the lower asymptote values.

x = "irt.pars" This method can be used to compute probabilities for the dichotomous items in an object of class "irt.pars". If x contains polytomous items, a warning will be displayed stating that probabilities will be computed for the dichotomous items only. If x contains parameters for multiple groups, a list of "irt.prob" objects will be returned. The argument dimensions does not need to be included for this method.

x = "sep.pars" This method can be used to compute probabilities for the dichotomous items in an object of class sep.pars. If x contains polytomous items, a warning will be displayed stating that probabilities will be computed for the dichotomous items only. The argument dimensions does not need to be included for this method.

Note

The indices 0 and 1 are used in the column labels for incorrect and correct responses respectively (e.g. the label d1.0 indicates the incorrect response column for item 1, d1.1 indicates the correct response column for the same item). If incorrect = FALSE, all column labels will end with .1

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Adams, R. J., Wilson, M., & Wang, W. (1997). The multidimensional random coefficients multinomial logit model. *Applied Psychological Measurement*, *21*(1), 1-23.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability In F. M. Lord & M. R. Novick (Eds.) *Statistical Theories of Mental Test Scores.* Reading, MA: Addison-Wesley.

Rasch, G. (1960). *Probabilistic Models for Some Intelligence and Attainment Tests* Copenhagen, Denmark: Danish Institute for Educational Research.

Reckase, M. D. (1985). The difficulty of items that measure more than one ability. *Applied Psychological Measurement*, *9*(4), 401-412.

Reckase, M. D. (1997). A linear logistic multidimensional model for dichotomous item response data. In W. J. van der Linden & R. K. Hambleton (Eds.) *Handbook of Modern Item Response Theory* (pp. 271-286). New York: Springer-Verlag.

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL http://www.jstatsoft.org/v35/i12/

See Also

mixed: compute probabilities for mixed-format items
plot: plot item characteristic/category curves
irt.prob, irt.pars, sep.pars: classes

Examples

```
###### Unidimensional Examples ######
## 1PL
# A single item with a difficulty at 1 logit
x <- drm(1)
plot(x)

# Three items with a difficulties at -2.2, -1.7, and 0.54 logits
# respectively. Assume a constant discrimination of 1 for each item
b <- c(-2.2,-.7,1.54)
x <- drm(b)
# Plot the expected probabilites
plot(x)

# Five items with a constant discrimination equal to 1.2
# Use the scaling constant for the normal metric
a <- rep(1.2,5)
b <- c(-2,-1,0,1,2)
pars <- list(a,b)
x <- drm(pars,D=1.7)
# Extract the expected probabilities
get.prob(x)
# Plot the expected probabilities
plot(x,combine=5,item.names="Items 1-5")


## 2PL
# Compute probabilities for five items at a theta value of 1.5 logits
# Print the model
a <- c(1,.8,.6,1.2,.9)
```

```
b <- c(-1.7,2.2,.4,-1,-.4)
pars <- cbind(a,b)
drm(pars,theta=1.5,print.mod=TRUE)


# Include a vector of zeros for the lower asymptote
# Compute probabilities for incorrect responses
a <- c(.63,1.14,.89,1.01,1.51,.58)
b <- c(-1.9,.08,1.6,-1.4,.5,-2.3)
c <- rep(0,6)
pars <- cbind(a,b,c)
x <- drm(pars,incorrect=TRUE)
plot(x)


## 3PL
a <- c(1,.8,.4,1.5,.9)
b <- c(-2,-1,0,1,2)
c <- c(.2,.25,.18,.2,.22)
pars <- list(a,b,c)
x <- drm(pars)
plot(x)

# Use theta values from -3 to 3 with an increment of 0.2
a <- c(.71,.96,.36,1.05,1.76,.64)
b <- c(-.16,1.18,2.03,1.09,.82,-1.56)
c <- c(.22,.17,.24,.08,.20,.13)
theta <- seq(-3,3,.2)
pars <- cbind(a,b,c)
x <- drm(pars,theta)
plot(x,combine=6,item.names="Items 1-6",auto.key=list(space="right"))

###### Multidimensional Examples ######
## M1PL
# A single item with a parameter related to item difficulty at 1 logit
x <- drm(1, dimensions=2)
plot(x)

# Three items with a difficulties at -2.2, -1.7, and 0.54 logits
# respectively. Assume a constant discrimination of 1 for each item
d <- c(-2.2,-.7,1.54)
x <- drm(d, dimensions=2)
plot(x, drape=TRUE)


## M2PL
# Items 27-30 from Reckase (1985)
a <- matrix(c(1.66,1.72,.69,.19,.88,1.12,.68,1.21),4,2,byrow=TRUE)
d <- c(-.38,-.68,-.91,-1.08)
pars <- list(a,d)
x <- drm(pars,dimensions=2)
plot(x, drape=TRUE, item.names=paste("Item",27:30))
```

```
# Create contourplots for these items
plot(x,type="contourplot",cuts=10)


## M3PL
# Single item from Reckase (1997, p. 274)
pars <- t(c(.8,1.4,-2,.2))
x <- drm(pars, dimensions=2)
plot(x, default.scales=list(arrows=FALSE),drape=TRUE)

# Compute the probabilities of an incorrect response
# for the Reckase (1997) item
x <- drm(pars, dimensions=2, incorrect=TRUE)
plot(x, screen=list(z=-40,x=-60), auto.key=list(space="right"))

# Four items from the included example for BMIRT (Yao, 2003)
# modeled using four dimensions
pars <- matrix(c(0.5038,2.1910,1.1317,0.2493,0.5240,0.1763,
0.2252,1.1999,0.5997,0.2087, 2.1730,0.4576,
0.2167,0.2487,1.4009,0.3865,-1.5270,0.3507,
2.3428,1.1530,0.3577,0.4240,-1.4971,0.3641),4,6,byrow=TRUE)
colnames(pars) <- c("a1","a2","a3","a4","d","c")
x <- drm(pars, dimensions=4, print.mod=TRUE)


## Not run:
# Plot the item response surfaces for item 4
plot(x,items=4,item.names="Item 4",drape=TRUE)

## End(Not run)
```

---

equate-methods   *IRT True Score and Observed Score Equating*

Description

This function conducts IRT true score and observed score equating for unidimensional single-format or mixed-format item parameters for two or more groups. This function supports all item response models available in plink with the exception of the multiple-choice model.

Usage

```
equate(x, method=c("TSE", "OSE"), true.scores, ts.low, base.grp=1, score=1,
  startval, weights1, weights2, syn.weights, exclude, max.tse.iter, ...)

## S4 method for signature 'list'
equate(x, method, true.scores, ts.low, base.grp, score, startval,
  weights1, weights2, syn.weights, exclude, max.tse.iter, ...)
```

```
## S4 method for signature 'irt.pars', 'ANY'
equate(x, method, true.scores, ts.low, base.grp, score, startval,
  weights1, weights2, syn.weights, exclude, max.tse.iter, ...)
```

Arguments

x	an object of class `irt.pars` with one or more groups, a list containing two or more `irt.pars` objects (e.g., when there are no common items), or the output from plink containing rescaled item parameters.
method	character vector identifying the equating method(s) to use. Values can include "TSE" and "OSE".
true.scores	numeric vector of true score values to be equated
ts.low	logical value. If TRUE, interpolate values for the equated true scores in the range of observed scores from one to the value below the lowest estimated true score (a rounded sum of guessing parameters)
base.grp	integer identifying the group for the base scale
score	if score = 1, score responses for the true-score equating method with zero for the lowest category and k-1 for the highest, k, category for each item. If score = 2, score responses with one for the lowest category and k for the highest, k, category for each item. A vector or list of scoring weights for each response category can be supplied, but this is only recommended for advanced users.
startval	integer starting value for the first value of true.score
weights1	list containing information about the theta values and weights to be used in the observed score equating for population 1. See below for more details.
weights2	list containing information about the theta values and weights to be used in the observed score equating for population 2. See below for more details.
syn.weights	vector of length two or a list containing vectors of length two with synthetic population weights to be used for each pair of tests for populations 1 and 2 respectively. If missing, weights of 0.5 will be used for both populations for all groups. If syn.weights is a list, there should be k-1 elements for k groups.
exclude	character vector or list identifying common items that should be excluded when estimating the linking constants. See below for more details.
max.tse.iter	maximum number of iterations to identify the theta value associated with each true score. The default is 50.
...	further arguments passed to or from other methods. See below for details.

Details

weights1 can be a list or a list of lists. The purpose of this object is to specify the theta values for population 1 to integrate over in the observed score equating as well as any weights associated with the theta values. The function `as.weight` can be used to facilitate the creation of this object. If weights1 is missing, the default is to use equally spaced theta values ranging from -4 to 4 with an increment of 0.05 and normal density weights for all groups.

To better understand the elements of weights1, let us assume for a moment that x has parameters for only two groups. In this instance, weights1 would be a single list with length two. The first

element should be a vector of theta values corresponding to points on the base scale. The second list element should be a vector of weights corresponding the theta values. If x contains more than two groups, a single `weights1` object can be supplied, and the same set of thetas and weights will be used for all adjacent groups. However, a separate list of theta values and weights for each adjacent group in x can be supplied.

The specification of `weights2` is the same as that for `weights1`, although the theta values and weights for this object correspond to theta values for population 2. This argument is only used when the synthetic weight associated with population 2 is greater than zero. If `weights2` is missing, the same theta values and weights used for `weights1` will be used for `weights2`.

For both equating methods, response probabilities are computed using the functions `drm`, `grm`, `gpcm`, and `nrm` for the associated models respectively. Various arguments from these functions can be passed to `equate`. Specifically, the argument `incorrect` can be passed to `drm` and `catprob` can be passed to `grm`. In the functions `drm`, `grm`, and `gpcm` there is an argument `D` for the value of a scaling constant. In `plink`, a single argument `D` can be passed that will be applied to all applicable models, or arguments `D.drm`, `D.grm`, and `D.gpcm` can be specified for each model respectively. If an argument is specified for `D` and, say `D.drm`, the values for `D.grm` and `D.gpcm` (if applicable) will be set equal to `D`. If only `D.drm` is specified, the values for `D.grm` and `D.gpcm` (if applicable) will be set to 1.

There are instances where certain items should not be included in the computation of total scores (e.g., when the common items correspond to an external anchor test or when using field test items) The `exclude` argument can be used to remove these items prior to conducting the equationg. `exclude` can be specified as a character vector or a list. In the former case, a single value "all.common" can be used to remove all common items or a vector of model names (i.e., "drm", "grm", "gpcm", "nrm", "mcm") can be supplied, indicating that any item on any test associated with the given model(s) would be excluded. If the argument is specified as a list, `exclude` should have as any elements as groups in x. Each list element can include model names and/or item numbers corresponding to the items on each test that should be excluded. If no items need to be excluded for a given group, the list element should be equal NA. For example, say we have two groups and we would like to exclude the GRM items and item 23 from the first group, we would specify `exclude` as `exclude <- list(c("grm",23),NA)`.

Value

Returns a matrix of equated true scores and/or a list of equated observed scores with associated marginal distributions or a list combining these two objects. The output for the observed-score equating also includes EAP scores and SDs for each of the observed scores (Thissen & Orlando, 2001).

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Kolen, M. J. (1981). Comparison of traditional and item response theory methods for equating tests. *Journal of Educational Measurement*, *18*(1), 1-11.

Kolen, M. J. & Brennan, R. L. (2004) *Test Equating, Scaling, and Linking* (2nd ed.). New York: Springer

Thissen, D. & Orlando, M. (2001) Item response theory for items scored in two categories. In D. Thissen & H. Wainer (Eds.) *Test Scoring* (p. 23 - 72). Hillsdale, NJ: Lawrence Erlbaum Associates.

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL http://www.jstatsoft.org/v35/i12/

Examples

```
# IRT true score and observed score examples from
# Kolen & Brennan (2004, ch. 6)
pm <- as.poly.mod(36)
x <- as.irt.pars(KB04$pars, KB04$common,
  cat=list(rep(2,36),rep(2,36)), poly.mod=list(pm,pm))
out <- plink(x, rescale="MS", base.grp=2, D=1.7, exclude=list(27,NA))

# Create the quadrature points and weights
wt <- as.weight(
  theta=c(-5.2086,-4.163,-3.1175,-2.072,-1.0269,0.0184,
    1.0635,2.109,3.1546,4.2001),
  weight=c(0.000101,0.00276,0.03021,0.142,0.3149,0.3158,
    0.1542,0.03596,0.003925,0.000186))

# Conduct the equating
equate(out,weights1=wt, synth.weights=c(1,0),D=1.7)

# Conduct true score equating for specific true scores
equate(out, true.scores=7:15, ts.low=FALSE, D=1.7)

# Exclude all common items (assume they correspond to an external anchor)
equate(out, D=1.7, exclude="all.common")


# Observed score equating for mixed-format tests
pm1 <- as.poly.mod(55,c("drm","gpcm","nrm"),dgn$items$group1)
pm2 <- as.poly.mod(55,c("drm","gpcm","nrm"),dgn$items$group2)
x <- as.irt.pars(dgn$pars,dgn$common,dgn$cat,list(pm1,pm2))
out <- plink(x, rescale="HB")
OSE <- equate(out, method="OSE", score=2)

# Display the equated scores
OSE[[1]]

# Multiple group equating
pars <- TK07$pars
common <- TK07$common
cat <- list(rep(2,26),rep(2,34),rep(2,37),rep(2,40),rep(2,41),rep(2,43))
pm1 <- as.poly.mod(26)
pm2 <- as.poly.mod(34)
pm3 <- as.poly.mod(37)
pm4 <- as.poly.mod(40)
pm5 <- as.poly.mod(41)
pm6 <- as.poly.mod(43)
pm <- list(pm1, pm2, pm3, pm4, pm5, pm6)
```

```
x <- as.irt.pars(pars, common, cat, pm, grp.names=paste("grade",3:8,sep=""))
out <- plink(x, rescale="SL")


# True score equating
equate(out, method="TSE")

# True score equating with the base group changed to 3
equate(out, method="TSE", base.grp=3)

# Observed score equating (These data are for non-equivalent groups, but
# this example is included to illustrate the multigroup capabilities)
OSE <- equate(out, method="OSE", base.grp=3)

# Display the equated scores for each group
OSE[[1]]
```

---

gpcm-methods   *Generalized Partial Credit Model Response Probabilities*

Description

This function computes the probability of responding in a specific category for one or more items
for a given set of theta values using the partial credit model, generalized partial credit model, or
multidimensional extension of these models, depending on the included item parameters and the
specified number of dimensions.

Usage

```
gpcm(x, cat, theta, dimensions = 1, D = 1, location = FALSE,
  print.mod = FALSE, items, information = FALSE, angle, ...)

## S4 method for signature 'matrix', 'numeric'
gpcm(x, cat, theta, dimensions, D, location, print.mod, items, information, angle, ...)

## S4 method for signature 'data.frame', 'numeric'
gpcm(x, cat, theta, dimensions, D, location, print.mod, items, information, angle, ...)

## S4 method for signature 'list', 'numeric'
gpcm(x, cat, theta, dimensions, D, location, print.mod, items, information, angle, ...)

## S4 method for signature 'irt.pars', 'ANY'
gpcm(x, cat, theta, dimensions, D, location, print.mod, items, information, angle, ...)

## S4 method for signature 'sep.pars', 'ANY'
gpcm(x, cat, theta, dimensions, D, location, print.mod, items, information, angle, ...)
```

Arguments

x	an R object containing item parameters
cat	vector identifying the number of response categories (not the number of step parameters) for each item.
theta	vector, matrix, or list of theta values for which probabilities will be computed. If theta is not specified, an equal interval range of values from -4 to 4 is used with an increment of 0.5. See details below for more information.
dimensions	number of modeled dimensions
D	scaling constant. The default value assumes that the parameters are already in the desired metric. If the parameters are in the logistic metric, they can be transformed to a normal metric by setting D = 1.7
location	if TRUE, the step parameters are deviations from a difficulty parameter
print.mod	if TRUE, print the model that was used (i.e., Partial Credit Model, Generalized Partial Credit Model, Multidimensional Partial Credit Model or Multidimensional Generalized Partial Credit Model)
items	numeric vector identifying the items for which probabilities should be computed
information	logical value. If TRUE compute item information. In the multidimensional case, information will be computed in the directions specified by angle or default angles of 0 - 90 in increments of 10 degrees.
angle	vector or matrix of angles between the dimension 1 axis and the corresponding axes for each of the other dimensions for which information will be computed. When there are more than two dimensions and angle is a vector, the same set of angles will be used relative to each of the corresponding axes.
...	further arguments passed to or from other methods

Details

theta can be specified as a vector, matrix, or list. For the unidimensional case, theta should be a vector. If a matrix or list of values is supplied, they will be converted to a single vector of theta values. For the multidimensional case, if a vector of values is supplied it will be assumed that this same set of values should be used for each dimension. Probabilities will be computed for each combination of theta values. Similarly, if a list is supplied, probabilities will be computed for each combination of theta values. In instances where probabilities are desired for specific combinations of theta values, a j x m matrix should be specified for j ability points and m dimensions where the columns are ordered from dimension 1 to m.

Value

Returns an object of class `irt.prob`

Methods

In the following description, references to the partial credit model and generalized partial credit model should be thought of as encompassing both the unidimensional and multidimensional models.

x = "matrix", cat = "numeric" This method allows one to specify an n x k matrix for n items. The number of columns can vary depending on the model (partial credit or generalized partial credit model), number of dimensions, and whether a location parameter is included. Generally, the first m columns, for m dimensions, are for item slopes and the remaining columns are for step parameters.

   Slope Parameters: The partial credit model is typically specified with all slopes equal to 1. For this model it is unnecessary (although optional) to include ones in the first m columns. For slope values other than one (equal for all items) or for the generalized partial credit model, slope parameters should be included in the first m columns.

   Step/Step Deviation Parameters: Step parameters can be characterized in two ways: as the actual steps or deviations from an overall item difficulty (location). In the deviation scenario the location argument should be equal to TRUE. If column(s) are included for the slope parameters, the location parameters should be in the m+1 column; otherwise, they should be in the first column. The columns for the step/step deviation parameters will always follow the slope and/or location columns (or they may potentially start in the first column for the partial credit model with no location parameter).

   The number of step/step deviation parameters can vary for each item. In these instances, all cells with missing values should be filled with NAs. For example, for a unidimensional generalized partial credit model with no location parameter, if one item has five categories (four step parameters) and another item has three categories (two step parameters), there should be five columns. The first column includes the slope parameters and columns 2-5 include the step parameters. The values in the last two columns for the item with three categories should be NA.

x = "data.frame", cat = "numeric" See the method for x = "matrix"

x = "list", cat = "numeric" This method can include a list with one or two elements. Generally, the first element is for item slopes and the second is for step/step deviation parameters.

   Slope Parameters: For the partial credit model with all slopes equal to 1 it is unnecessary (although optional) to include a list element for the item slopes. If no slope values are included, the first element would contain the step/deviation step parameters. For slopes other than 1 (equal for all items) or for the generalized partial credit model, slope values should be included in the first list element. For the unidimensional case, these values should be a vector of length n or an n x 1 matrix for n items. For the multidimensional case, an n x m matrix of values for m dimensions should be supplied

   Step/Step Deviation Parameters: The step/step deviation parameters should be formatted as n x k matrix for n items. If the steps are deviations from a location parameter, the argument location should equal TRUE and the location parameters should be in the first column. The number of step/step deviation parameters can vary for each item. In these instances, all cells with missing values should be filled with NAs (See the example in the method for x = "matrix").

x = "irt.pars", cat = "ANY" This method can be used to compute probabilities for the gpcm items in an object of class "irt.pars". If x contains dichotomous items or items associated with another polytomous model, a warning will be displayed stating that probabilities will be computed for the gpcm items only. If x contains parameters for multiple groups, a list of "irt.prob" objects will be returned. The argument dimensions does not need to be included for this method.

x = "sep.pars", cat = "ANY" This method can be used to compute probabilities for the gpcm items in an object of class sep.pars. If x contains dichotomous items or items associated

with another polytomous model, a warning will be displayed stating that probabilities will be computed for the gpcm items only. The argument `dimensions` does not need to be included for this method.

Note

The determination of the model (partial credit or generalized partial credit) is based on the number of non-NA columns for each item in `x` and the corresponding values in `cat`.

Author(s)

Jonathan P. Weeks `<weeksjp@gmail.com>`

References

Adams, R. J., Wilson, M., & Wang, W. (1997). The multidimensional random coefficients multinomial logit model. *Applied Psychological Measurement*, *21*(1), 1-23.

Embretson, S. E., & Reise, S. P. (2000). *Item Response Theory for Psychologists.* Mahwah, New Jersey: Lawrence Erlbaum Associates.

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

Masters, G. N. & Wright, B. D. (1996) The partial credit model. In W. J. van der Linden & Hambleton, R. K. (Eds.) *Handbook of Modern Item Response Theory* (pp. 101-121). New York: Springer-Verlag.

Muraki, E. (1992) A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement*, *16*(2), 159-176.

Muraki, E. (1996) A generalized partial credit model. In W. J. van der Linden & Hambleton, R. K. (Eds.) *Handbook of Modern Item Response Theory* (pp. 153-164). New York: Springer-Verlag.

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL http://www.jstatsoft.org/v35/i12/

Yao, L. (2003). BMIRT: Bayesian multivariate item response theory [Computer Program]. Monterey, CA: CTB/McGraw-Hill.

Yao, L., & Schwarz, R. D. (2006). A multidimensional partial credit model with associated item and test statistics: An application to mixed-format tests. *Applied Psychological Measurement*, *30*(6), 469-492.

See Also

mixed: compute probabilities for mixed-format items
plot: plot item characteristic/category curves
irt.prob, irt.pars, sep.pars: classes

Examples

```
###### Unidimensional Examples ######
## Partial Credit Model
## Item parameters from Embretson & Reise (2000, p. 108) item 5
b <- t(c(-2.519,-.063,.17,2.055))
```

```
x <- gpcm(b,5)
plot(x)

## Generalized Partial Credit Model
## Item parameters from Embretson & Reise (2000, p. 112) items 5-7
a <- c(.683,1.073,.583)
b <- matrix(c(-3.513,-.041,.182,NA,-.873,.358,-.226,
  1.547,-4.493,-.004,NA,NA),3,4,byrow=TRUE)
pars <- cbind(a,b) # Does not include a location parameter
rownames(pars) <- paste("Item",5:7,sep="")
colnames(pars) <- c("a",paste("b",1:4,sep=""))
cat <- c(4,5,3)
x <- gpcm(pars,cat,seq(-3,3,.05))
plot(x)

## Item parameters from Muraki (1996, p. 154)
a <- c(1,.5)
b <- matrix(c(.25,-1.75,1.75,.75,-1.25,1.25),2,3,byrow=TRUE)
pars <- cbind(a,b) # Include a location parameter
rownames(pars) <- paste("Item",1:2,sep="")
colnames(pars) <- c("a","b",paste("d",1:2,sep=""))
cat <- c(3,3)
x <- gpcm(pars,cat,location=TRUE,print.mod=TRUE, D=1.7)
# Plot category curves for two items
matplot(x@prob$theta,x@prob[,2:4],xlab="Theta",ylab="Probability",
  ylim=c(0,1),lty=1,type="l",col="black")
par(new=TRUE)
matplot(x@prob$theta,x@prob[,5:7],xlab="Theta",ylab="Probability",
  ylim=c(0,1),lty=3,type="l",col="black")


###### Multidimensional Examples ######
## Multidimensional Partial Credit Model
pars <- matrix(c(2.4207,0.245,-1.1041,NA,
2.173,-0.4576,NA,NA,
2.1103,-0.8227,.4504,NA,
3.2023,1.0251,-.7837,-1.3062),4,4,byrow=TRUE)
cat <- c(4,3,4,5)
x <- gpcm(pars,cat,dimensions=2,print.mod=TRUE)
# plot combined item category surfaces
# The screen argument adjusts the orientation of the axes
plot(x,screen=list(z=-60,x=-70))

## Multidimensional Generalized Partial Credit Model
a <- matrix(c(.873, .226, .516, .380, .613, .286 ),3,2,byrow=TRUE)
b <- matrix(c(2.255, 1.334, -.503, -2.051, -3.082,
1.917, 1.074, -.497, -1.521, -2.589,
1.624, .994, -.656, -1.978, NA),3,5,byrow=TRUE)
pars <- cbind(a,b)
cat <- c(6,6,5)
x <- gpcm(pars,cat,dimensions=2,print.mod=TRUE)

# plot combined item category surfaces
```

```
plot(x,screen=list(z=-40,x=-60), auto.key=list(space="right"))

# plot separated item category surfaces for item two
plot(x,items=2,separate=TRUE,drape=TRUE,panels=1)

# Compute response probabilities for a single three-category item with
# three dimensions. Plot the response surfaces for the first two
# dimensions conditional on each theta value on the third dimension
pars <- matrix(c(1.1999,0.5997,0.8087,2.1730,-1.4576),1,5)
x <- gpcm(pars,3,dimensions=3,theta=-4:4)
plot(x, screen=list(z=-30,x=-60))
```

---

grm-methods   *Graded Response Model Probabilities*

Description

This function computes the cumulative probability of responding within or above a certain category or the probability of responding in a specific category for one or more items for a given set of theta values using the graded response model or multidimensional graded response model.

Usage

```
grm(x, cat, theta, dimensions = 1, catprob = FALSE, D = 1,
  location = FALSE, items, information = FALSE, angle, ...)

## S4 method for signature 'matrix', 'numeric'
grm(x, cat, theta, dimensions, catprob, D, location, items, information, angle, ...)

## S4 method for signature 'data.frame', 'numeric'
grm(x, cat, theta, dimensions, catprob, D, location, items, information, angle, ...)

## S4 method for signature 'list', 'numeric'
grm(x, cat, theta, dimensions, catprob, D, location, items, information, angle, ...)

## S4 method for signature 'irt.pars', 'ANY'
grm(x, cat, theta, dimensions, catprob, D, location, items, information, angle, ...)

## S4 method for signature 'sep.pars', 'ANY'
grm(x, cat, theta, dimensions, catprob, D, location, items, information, angle, ...)
```

Arguments

x	ans R object containing item parameters.
cat	vector identifying the number of response categories (not the number of threshold parameters) for each item.

theta	vector, matrix, or list of theta values for which probabilities will be computed. If theta is not specified, an equal interval range of values from -4 to 4 is used with an increment of 0.5. See details below for more information.
dimensions	number of modeled dimensions
catprob	if TRUE, compute category probabilities instead of cumulative probabilities
D	scaling constant. The default value assumes that the parameters are already in the desired metric. If the parameters are in the logistic metric, they can be transformed to a normal metric by setting D = 1.7
location	if TRUE, the step parameters are deviations from a difficulty parameter
items	numeric vector identifying the items for which probabilities should be computed
information	logical value. If TRUE compute item information. In the multidimensional case, information will be computed in the directions specified by angle or default angles of 0 - 90 in increments of 10 degrees.
angle	vector or matrix of angles between the dimension 1 axis and the corresponding axes for each of the other dimensions for which information will be computed. When there are more than two dimensions and angle is a vector, the same set of angles will be used relative to each of the corresponding axes.
...	further arguments passed to or from other methods

Details

theta can be specified as a vector, matrix, or list. For the unidimensional case, theta should be a vector. If a matrix or list of values is supplied, they will be converted to a single vector of theta values. For the multidimensional case, if a vector of values is supplied it will be assumed that this same set of values should be used for each dimension. Probabilities will be computed for each combination of theta values. Similarly, if a list is supplied, probabilities will be computed for each combination of theta values. In instances where probabilities are desired for specific combinations of theta values, a j x m matrix should be specified for j ability points and m dimensions where the columns are ordered from dimension 1 to m.

Value

Returns an object of class `irt.prob`

Methods

x = "matrix", cat = "numeric" This method allows one to specify an n x k matrix for n items. The number of columns can vary depending on the number of dimensions and whether a location parameter is included. The first m columns, for m dimensions, are for item slopes and the remaining columns are for the threshold/threshold deviation parameters.

Threshold parameters can be characterized in two ways: the actual thresholds or deviations from an overall item difficulty (location). In the deviation scenario the location argument should be TRUE and the location parameters should be in the m+1 column. The columns for the threshold/threshold deviation parameters will always follow the slope column(s) and, if applicable, the location column. The number of threshold/threshold deviation parameters can vary for each item. In these instances, all cells with missing values should be filled with NAs. For example, for a unidimensional model with no location parameter, if one item has five

categories (four threshold parameters) and another item has three categories (two threshold parameters), there should be five columns. The first column includes the slope parameters and columns 2-5 include the threshold parameters. The values in the last two columns for the item with three categories should be NA.

x = "data.frame", cat = "numeric" See the method for x = "matrix"

x = "list", cat = "numeric" This method is for a list with two elements. The first list element is for item slopes and the second for the threshold/threshold deviation parameters. For the unidimensional case, the slope values should be a vector of length n or an n x 1 matrix for n items. For the multidimensional case, the slopes should be specified in an n x m matrix. For both the unidimensional and multidimensional cases, the threshold/threshold deviation parameters should be formatted as an n x k matrix. If the thresholds are deviations from a location parameter, the argument location should be TRUE and the location parameters should be in the first column. The number of threshold/threshold deviation parameters can vary for each item. In these instances, all cells with missing values should be filled with NAs (See the example in the method for x = "matrix").

x = "irt.pars", cat = "ANY" This method can be used to compute probabilities for the grm items in an object of class "irt.pars". If x contains dichotomous items or items associated with another polytomous model, a warning will be displayed stating that probabilities will be computed for the grm items only. If x contains parameters for multiple groups, a list of "irt.prob" objects will be returned. The argument dimensions does not need to be included for this method.

x = "sep.pars", cat = "ANY" This method can be used to compute probabilities for the grm items in an object of class sep.pars. If x contains dichotomous items or items associated with another polytomous model, a warning will be displayed stating that probabilities will be computed for the grm items only. The argument dimensions does not need to be included for this method.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Embretson, S. E., & Reise, S. P. (2000) *Item Response Theory for Psychologists.* Mahwah, New Jersey: Lawrence Erlbaum Associates.

Kolen, M. J. & Brennan, R. L. (2004) *Test Equating, Scaling, and Linking (2nd ed.).* New York: Springer.

Muraki, E., & Carlson, J. E. (1995). Full-information factor analysis for polytomous item responses. *Applied Psychological Measurement*, *19*(1), 73-90.

Samejima, F. (1969) Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph*, No. 17.

Samejima, F. (1996) The graded response model. In W. J. van der Linden & Hambleton, R. K. (Eds.) *Handbook of Modern Item Response Theory* (pp. 85-100). New York: Springer-Verlag.

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL http://www.jstatsoft.org/v35/i12/

See Also

mixed: compute probabilities for mixed-format items
plot: plot item characteristic/category curves
irt.prob, irt.pars, sep.pars: classes

Examples

```
###### Unidimensional Examples ######
## Cumulative probabilities
## Item parameters from Embretson & Reise (2000, p. 101) items 9-11
# Includes a location parameter
a <- c(2.09,1.18,1.69)
b <- matrix(c(-1.07,-1.03,.39,.86,1.78,1.85,-.87,-.39,.31,NA,
  -1.4,-.42,NA,NA,-1.74),3,5)
pars <- cbind(a,b)
rownames(pars) <- paste("Item",9:11,sep=" ")
colnames(pars) <- c("a","b",paste("c",1:4,sep=""))
cat <- c(3,4,5)
x <- grm(pars,cat,location=TRUE)
plot(x)

## Item parameters from Kolen & Brennan (2004, p. 210)
# Use theta values from -3 to 3 with an increment of 0.5
pars <- t(c(1.2,-.5,.6,1.1,1.3))
x <- grm(pars,5,theta=seq(-3,3,.05))
plot(x,item.lab=FALSE)


## Category probabilities
## Single item
pars <- t(c(1.2,-.5,.6,1.1,1.3))
x <- grm(pars,5,seq(-3,3,.05),catprob=TRUE)
plot(x,item.lab=FALSE)

# Items without location parameter
a <- c(2.09,1.18,1.69)
b <- matrix(c(-1.93,-2.81,-1.46,-.2,-.64,.08,NA,.37,.81,NA,NA,2.13),3,4)
pars <- cbind(a,b)
rownames(pars) <- paste("Item",9:11,sep=" ")
colnames(pars) <- c("a",paste("b",1:4,sep=""))
cat <- c(3,4,5)
x <- grm(pars,cat,catprob=TRUE)
plot(x)

## Create sep.pars object then compute category probabilities
a <- c(2.09,1.18,1.69)
b <- matrix(c(-1.93,-2.81,-1.46,-.2,-.64,.08,NA,.37,.81,NA,NA,2.13),3,4)
pars <- cbind(a,b)
cat <- c(3,4,5)
pm <- as.poly.mod(3,"grm")
out <- sep.pars(pars,cat,pm)
x <- grm(out,catprob=TRUE)
```

```
plot(x)


###### Multidimensional Examples ######
## Cumulative probabilities
a <- matrix(c(.873, .226, .516, .380, .613, .286 ),3,2,byrow=TRUE)
d <- matrix(c(2.255, 1.334, -.503, -2.051, -3.082,
1.917, 1.074, -.497, -1.521, -2.589,
1.624, .994, -.656, -1.978, NA),3,5,byrow=TRUE)
pars <- cbind(a,d)
cat <- c(6,6,5)
x <- grm(pars,cat,dimensions=2)
plot(x)

# Plot separated response surfaces
plot(x, separate=TRUE, drape=TRUE)

## Category Probabilities
## Use {pars} an {cat} from the example above
x <- grm(pars,cat,dimensions=2, catprob=TRUE)

# plot combined item category surfaces
# The screen argument adjusts the orientation of the axes
plot(x,screen=list(z=-40,x=-60), auto.key=list(space="right"))
```

---

irt.pars-class   *Class "irt.pars"*

Description

The formal S4 class for irt.pars. This class contains the item parameters and characteristics of the
item parameters for one or more groups. When parameters are included for two or more groups, the
common items between the different groups are also included.

Details

Objects of class `irt.pars` contain all the information necessary to produce expected response prob-
abilities, linking constants (when data for two or more groups are included), and to plot item re-
sponse curves/surfaces, vector plots, and parameter comparison plots.

Objects from the Class

Objects can be created by calls of the form new("irt.pars", ...), but this is not encouraged. Use
the function as.irt.pars instead.

Slots

pars: matrix of item parameters for a single group or a list of matrices containing item parameters
    for two or more groups

cat: vector with length equal to the number of items, identifying the number of response categories for each item for a single group or a list of response category vectors for two or more groups

poly.mod: a `poly.mod` object for one group or a list of `poly.mod` objects identifying the items associated with each IRT model (see class `poly.mod` for more information)

common: a j x 2 matrix for j common items or a list of matrices identifying the common items between pairs of item parameters

location: logical vector identifying whether a given set of item parameters (i.e., for a given group) includes a location parameter

groups: the number of groups (i.e. sets of item parameters)

dimensions: numeric vector identifying the number of modeled dimensions

Extends

Class `poly.mod`, directly. Class `list.poly`, by class poly.mod, distance 2.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

See Also

`as.irt.pars`, `as.poly.mod`

---

irt.prob-class   *Class "irt.prob"*

Description

The formal S4 class for irt.prob. This class contains the expected probabilities of responding in a given category for a set of items and theta values under the specified IRT models. The class also includes characteristics of the items.

Objects from the Class

Objects can be created by calls of the form new("irt.prob", ...), but this is not encouraged. Use one of the functions drm, gpcm, grm, mcm, nrm, or mixed instead.

Slots

prob: data.frame of item probabilities with n rows and j+m columns for n theta values and j items (the first m column contains theta values for m dimensions)

info: data.frame of item information

p.cat: vector identifying the number of categories for each item for which probabilities were computed

mod.lab: character vector of labels for the model(s).

dimensions: numeric value identifying the number of modeled dimensions

D: numeric vector identifying scaling constants for drm, grm, and gpcm

pars: list of the item parameters used to compute the probabilities

model: character vector identifying all the models used to compute the probabilities in prob. The only acceptable models are drm, gpcm, grm, mcm, and nrm (see class poly.mod for more information).

items: list with the same length as model, where each element identifies the items associated with the model(s) specified in model.

Extends

Class poly.mod, directly.
Class list.poly, by class poly.mod, distance 2.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

See Also

These models provide information on both unidimensional and multidimensional formulations
drm: for dichotomous response models (1PL, 2PL, and 3PL)
gpcm: for the partial credit/generalized partial credit model
grm: for the graded response model
mcm: for the multiple-choice model
nrm: for the nominal response model
mixed: for mixed-format items
irt.pars

---

is.irt.pars *Identify Class*

Description

These functions test whether a given object is of a specified class

Usage

```
is.irt.pars(x)
is.sep.pars(x)
is.poly.mod(x)
is.irt.prob(x)
is.link(x)
```

Arguments

x   an R object

Value

Returns a logical value

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

See Also

[irt.pars](), [sep.pars](), [poly.mod](), [irt.prob](), [link]()

KB04 *Kolen - Brennan (2004)*

Description

This (unidimensional) dataset includes three parameter logistic model (3PL) item parameter estimates for two groups. The Form X items are from a "new" form and the Form Y items are from an "old" form. The data are listed in Table 6.5 in *Test Equating, Scaling, and Linking* (2nd ed.). There are 12 common items between the two forms.

Usage

KB04

Format

A list of length two. The first element pars is a list of length two. Each of the list elements in pars is a 36 x 3 matrix of item parameters. Element one is for Form X, and element two is for Form Y.

The second list element in KB04 is a matrix identifying the common items between the two sets of parameters in pars. That is, the first column in common identifies the rows in pars$form.x that are common items. The second column identifies the corresponding set of common items in pars$form.y.

Note

In the book, the authors transform the Form X parameters to the Form Y scale.

Source

Kolen, M. J., & Brennan, R. L. (2004) *Test Equating, Scaling, and Linking.* (2nd ed.), p. 192 New York: Springer

---

link-class   *Class "link"*

Description

The formal S4 class for link. This class compiles the linking constants, item parameter descriptive statistics, and iteration/objective/convergence information for the moment methods and characteristic curve methods from a separate calibration linking procedure. It is also able to store rescaled item and ability parameters.

Objects from the Class

Objects can be created by calls of the form new("link", ...), but this is not encouraged. Use the function [plink](#) instead.

Slots

constants: list of linking methods where the list elements include the linking constants. For unidimensional models, each element is a vector of length two containing a slope and intercept to adjust the SD and mean respectively. For multidimensional models, each element is a list with a matrix (or set of matrices) to resolve rotational indeterminacy and scale, and a vector of translation constants to adjust the mean of each dimension.

descriptives: data.frame or list containing the item parameter descriptive statistics

iterations: vector of the number of optimization iterations for the Haebara and Stocking-Lord methods

objective: vector of criterion values for HBcrit and SLcrit at the point of convergence for the Haebara and Stocking-Lord methods

convergence: character vector identifying the type of convergence reached under the Haebara or Stocking-Lord optimizations.

base.grp: numeric value indicating the base group for the calibration

n: vector identifying the total number of common items, the number of dichotomous common items, and the number of polytomous common items

grp.names: character vector of group names

mod.lab: character vector identifying the dichotomous and/or polytomous models used to model the item responses

dilation: character value identifying the dilation approach used when estimating multidimensional linking constants

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

See Also

[plink](#)

---

link.con   *Extract Output Information*

Description

These functions extract information from an object of class irt.pars, typically created using plink, or an object of class irt.prob containing expected probabilities.

Usage

```
link.con(x, method = "ALL")
link.pars(x)
link.ability(x)
get.prob(x)
```

Arguments

x   an R object

method   character vector identifying the linking methods for which constants should be
                returned. The only acceptable values are ALL, MM, MS, HB, SL, and RM. See be-
                low for details.

Details

link.con extracts the linking constants, link.pars extracts the rescaled item parameters (if present),
and link.ability extracts the rescaled ability estimates (if present) get.prob extracts expected
probabilities for an irt.prob object

For link.con, the following values can be included for method. For multidimensional constants,
although matrices A, K, and T are included in the object of class "link" for the Haebara and Stocking-
Lord methods, only the matrix A will be returned for these methods.

ALL Returns the constants for all linking methods

MM: Mean/Mean

MS: Mean/Sigma

HB: Haebara

SL: Stocking-Lord

RM: Reckase-Martineau (for multidimensional constants only)

Value

Returns a matrix or list of linking constants

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

See Also

link, plink

---

list.poly-class   *Class Combinations*

Description

These classes are combinations of classes

Details

list.poly: combines class `list` and class [`poly.mod`](poly.mod)

list.dat: combines class `list` and class `data.frame`

list.mat: combines class `list` and class `matrix`

list.num: combines class `list` and class `numeric`

num.null: combines class `numeric` and class `NULL`

list.null: combines class `list` and class `NULL`

pars.null: combines class `irt.pars` and class `NULL`

dat.null: combines class `data.frame` and class `NULL`

Objects from the Class

A virtual Class: No objects may be created from it.

Author(s)

Jonathan P. Weeks `<weeksjp@gmail.com>`

---

mcm-methods   *Multiple-Choice Model Response Probabilities*

Description

This function computes the probability of responding in a specific category for one or more items for a given set of theta values using the multiple-choice model or multidimensional multiple-choice model.

Usage

```
mcm(x, cat, theta, dimensions = 1, items, information = FALSE, angle, ...)

## S4 method for signature 'matrix', 'numeric'
mcm(x, cat, theta, dimensions, items, information, angle, ...)

## S4 method for signature 'data.frame', 'numeric'
mcm(x, cat, theta, dimensions, items, information, angle, ...)

## S4 method for signature 'list', 'numeric'
mcm(x, cat, theta, dimensions, items, information, angle, ...)

## S4 method for signature 'irt.pars', 'ANY'
mcm(x, cat, theta, dimensions, items, information, angle, ...)

## S4 method for signature 'sep.pars', 'ANY'
mcm(x, cat, theta, dimensions, items, information, angle, ...)
```

Arguments

x	an R object containing item parameters
cat	vector identifying the number of response categories plus one for each item (the additional category is for 'do not know')
theta	vector, matrix, or list of theta values for which probabilities will be computed. If theta is not specified, an equal interval range of values from -4 to 4 is used with an increment of 0.5. See details below for more information.
dimensions	number of modeled dimensions
items	numeric vector identifying the items for which probabilities should be computed
information	logical value. If TRUE compute item information. In the multidimensional case, information will be computed in the directions specified by angle or default angles of 0 - 90 in increments of 10 degrees.
angle	vector or matrix of angles between the dimension 1 axis and the corresponding axes for each of the other dimensions for which information will be computed. When there are more than two dimensions and angle is a vector, the same set of angles will be used relative to each of the corresponding axes.
...	further arguments passed to or from other methods

Details

theta can be specified as a vector, matrix, or list. For the unidimensional case, theta should be a vector. If a matrix or list of values is supplied, they will be converted to a single vector of theta values. For the multidimensional case, if a vector of values is supplied it will be assumed that this same set of values should be used for each dimension. Probabilities will be computed for each combination of theta values. Similarly, if a list is supplied, probabilities will be computed for each combination of theta values. In instances where probabilities are desired for specific combinations of theta values, a j x m matrix should be specified for j ability points and m dimensions where the columns are ordered from dimension 1 to m.

Value

Returns an object of class `irt.prob`

Methods

x = "matrix", cat = "numeric" This method allows one to specify an n x h matrix for n items and h equal to [m x 2k + (k-1)] where m is the number of modeled dimensions and k is equal to the maximum number of response categories (including the 'do not know' category) across items. The first (m x k) columns are for category slope parameters, the next block of (m x k) columns are for category difficulty parameters, and the remaining columns are for the lower asymptote (guessing) parameters. For any items with fewer categories than the maximum, the remaining cells in each block of (m x k) columns or the last k columns should be NA.

Unidimensional Specification: Say we have one item with four actual response categories and one item with five response categories. There will be 17 columns. The first six columns are for the category slope parameters. The first column should contain the parameters for the 'do not know' category. Column six for the four category item should be NA. The next six columns (7-12) are for category difficulty parameters. The first column of this subset of columns (column 7) should contain the category difficulties for the 'do not know' category. Similar to the block of columns containing the slopes, the last column in this subset of columns (column 12) for the four category item should be NA. The remaining five columns are for the lower asymptote (guessing) parameters. The last column for the four category item would be NA.

Multidimensional Specification: In the multidimensional case, the columns for the slope and difficulty parameters should be grouped first by dimension and then by category. Using the same example for the two items with two dimensions there will be 29 columns. The first five columns for the four category item would include the slope parameters associated with the first dimension for the 'do not know' category and each of the four actual categories respectively. Columns 11-12 would be NA. Columns 13-17 would include the category difficulties associated with the first dimension (again the parameters for the 'do not know' category should be in the first column of this block of columns) and columns 23-24 would be NA. The remaining five columns are for the lower asymptote (guessing) parameters. The last column for the four category item would be NA.

x = "data.frame", cat = "numeric" See the method for x = "matrix"

x = "list", cat = "numeric" This method is for a list with three elements. The first element is an n x (m x k) matrix of category slope values for n items, m dimensions, and k equal to the maximum number of response categories across items (including the 'do not know' category). The second list element is an n x (m x k) matrix of category difficulty parameters and the last element is an n x (k-1) matrix of lower asymptote (guessing) parameters. For any list element, for items with fewer categories than the maximum, the remaining cells in the rows should be NA (see the examples for method x = "matrix" for specification details).

x = "irt.pars", cat = "ANY" This method can be used to compute probabilities for the mcm items in an object of class `"irt.pars"`. If x contains dichotomous items or items associated with another polytomous model, a warning will be displayed stating that probabilities will be computed for the mcm items only. If x contains parameters for multiple groups, a list of `"irt.prob"` objects will be returned. The argument `dimensions` does not need to be included for this method.

x = "sep.pars", cat = "ANY" This method can be used to compute probabilities for the mcm
items in an object of class sep.pars. If x contains dichotomous items or items associated
with another polytomous model, a warning will be displayed stating that probabilities will be
computed for the mcm items only. The argument dimensions does not need to be included
for this method.

Note

No multidimensional extension of the multiple-choice model has officially been presented in the
literature; however, this model is consistent with how the 3PL and nominal response model were
extended to the multidimensional context. Additionally, item information is not currently available
for the multiple-choice model even though the arguments are included in the function. Information
will be available in a later release.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Bolt, D. M. & Johnson, T. J. (in press) Applications of a MIRT model to self-report measures: Ad-
dressing score bias and DIF due to individual differences in response style. *Applied Psychological
Measurement*.

Thissen, D., & Steinberg, L. (1984). A response model for multiple choice items. *Psychometrika*,
49(4), 501-519.

Thissen, D., & Steinberg, L. (1996) A response model for multiple choice items. In W.J. van
der Linden & Hambleton, R. K. (Eds.) *Handbook of Modern Item Response Theory*. New York:
Springer-Verlag

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods.
Journal of Statistical Software, **35(12)**, 1–33. URL http://www.jstatsoft.org/v35/i12/

See Also

mixed: compute probabilities for mixed-format items
plot: plot item characteristic/category curves
irt.prob, irt.pars, sep.pars: classes

Examples

```
###### Unidimensional Examples ######
## Item parameters from Thissen & Steinberg (1984, p. 510)
## Items R,S,T,U for the whole test
a <- matrix(c(-1.7, -1, 1.1, .3, 1.9,
  -2.1, -.6, 1.2, 2.3, -.8,
  -1.3, -.9, -.2, 1.9, .5,
  -1.9, -.5, 0, -.6, 1.9),4,5,byrow=TRUE)
c <- matrix(c(.3, -2.3, 2.4, -2.5, 2.1,
  2.1, .05, -3, -.6, 1,
  -.9, -2.5, -.1, 1.8, 1.6,
  -.1, -2, .5, .8, .8),4,5,byrow=TRUE)
```

```
d <- matrix(c(.25, .25, .25, .25,
  .2, .2, .4, .2,
  .2, .2, .4, .2,
  .25, .25, .25, .25),
  4,4,byrow=TRUE)
pars <- cbind(a,c,d)
x <- mcm(pars, rep(5,4))
plot(x,item.names=paste("Item",c("R","S","T","U")),
  auto.key=list(space="right"))

## Item parameters from Thissen & Steinberg (1984, p. 511)
## Items W,X,Y,Z for the
pars <- vector("list",3)
pars[[1]] <- matrix(c(-2.3, -.2, 2, .9, -.3,
  -.8, .6, -.5, 1.1, -.4,
  -.5, -.2, 2, -1.2, 0,
  -1.5, -.7, -.2, .1, 2.3),4,5,byrow=TRUE)
pars[[2]] <- matrix(c(.5, .7, -.5, -1.9, 1.1,
  1.6, -2.8, 1.5, 0, -.3,
  -.3, .7, -1, .7, 0,
  .4, .4, -.5, .5, -.8),4,5,byrow=TRUE)
pars[[3]] <- matrix(c(.2, .4, .2, .2,
  .2, .2, .4, .2,
  .2, .4, .2, .2,
  .2, .2, .2, .4), 4,4,byrow=TRUE)
x <- mcm(pars, rep(5,4))
plot(x,item.names=paste("Item",c("W","X","Y","Z")),
  auto.key=list(space="right"))


###### Multidimensional Example ######
## Discrimination and category parameters from Bolt & Johnson (in press)
pars <- matrix(c(-1.28, -1.029, -0.537, 0.015, 0.519, 0.969, 1.343,
1.473, -0.585, -0.561, -0.445, -0.741, -0.584, 1.444,
0.29, 0.01, 0.04, 0.34, 0, -0.04, -0.63,
0.01, 0.09, 0.09, 0.28, 0.22, 0.31),1,27)
x <- mcm(pars, cat=7, dimensions=2)
# Plot separated surfaces
plot(x,separate=TRUE,drape=TRUE)
```

---

md.mixed   *Multidimensional Mixed-Format Tests*

Description

This dataset includes multidimensional 3PL and multidimensional generalized partial credit model item parameters for two groups modeled with four dimensions. The item parameters were estimated based on item responses provided with BMIRT (Yao, 2008).

Usage

```
md.mixed
```

Format

A list of length two. There are two matrices in the first list element that correspond to form.x and
form.y tests respectively. The second list element identifies the number of response categories for
each item. All of the items are common items. There is no location parameter for the polytomous
items.

Source

Yao, L. (2008). BMIRT: Bayesian multivariate item response theory [Computer Program]. Mon-
terey, CA: CTB/McGraw-Hill.

---

mixed-methods   *Mixed-Format Response Probabilities*

Description

This function computes the probability of responding in a specific category for one or more items
for a given set of theta values when the items are from a mixed-format test.

Usage

```
mixed(x, cat, poly.mod, theta, dimensions = 1, items, information = FALSE, angle, ...)

## S4 method for signature 'numeric', 'numeric'
mixed(x, cat, poly.mod, theta, dimensions, items, information, angle, ...)

## S4 method for signature 'matrix', 'numeric'
mixed(x, cat, poly.mod, theta, dimensions, items, information, angle, ...)

## S4 method for signature 'data.frame', 'numeric'
mixed(x, cat, poly.mod, theta, dimensions, items, information, angle, ...)

## S4 method for signature 'list', 'numeric'
mixed(x, cat, poly.mod, theta, dimensions, items, information, angle, ...)

## S4 method for signature 'irt.pars', 'ANY'
mixed(x, cat, poly.mod, theta, dimensions, items, information, angle, ...)

## S4 method for signature 'sep.pars', 'ANY'
mixed(x, cat, poly.mod, theta, dimensions, items, information, angle, ...)
```

Arguments

x	an R object containing item parameters
cat	vector identifying the number of response categories for each item. If multiple-choice model items are included, cat for these items should equal the number of response categories plus one (the additional category is for 'do not know')
poly.mod	object of class poly.mod identifying the items associated with each IRT model
theta	vector, matrix, or list of theta values for which probabilities will be computed. If theta is not specified, an equal interval range of values from -4 to 4 is used with an increment of 0.5. See details below for more information.
dimensions	number of modeled dimensions
items	numeric vector identifying the items for which probabilities should be computed
information	logical value. If TRUE compute item information. In the multidimensional case, information will be computed in the directions specified by angle or default angles of 0 - 90 in increments of 10 degrees.
angle	vector or matrix of angles between the dimension 1 axis and the corresponding axes for each of the other dimensions for which information will be computed. When there are more than two dimensions and angle is a vector, the same set of angles will be used relative to each of the corresponding axes.
...	further arguments passed to or from other methods. See details below.

Details

The item parameters supplied to this method can be associated with a single IRT model or multiple models. When the parameters are tied to only one model, the format of x (for either unidimensional or multidimensional models) should follow the conventions in drm for dichotomous response models (i.e. 1PL, 2PL, 3PL), gpcm for the partial credit model and generalized partial credit model, grm for the graded response model, mcm for the multiple-choice model, and nrm for the nominal response model. When the parameters are associated with two or more models, the parameters should be combined. See as.irt.pars or for more details on how the parameters from different models can be combined. Additional arguments for the above models can be passed to this method as well.

theta can be specified as a vector, matrix, or list. For the unidimensional case, theta should be a vector. If a matrix or list of values is supplied, they will be converted to a single vector of theta values. For the multidimensional case, if a vector of values is supplied it will be assumed that this same set of values should be used for each dimension. Probabilities will be computed for each combination of theta values. Similarly, if a list is supplied, probabilities will be computed for each combination of theta values. In instances where probabilities are desired for specific combinations of theta values, a j x m matrix should be specified for j ability points and m dimensions where the columns are ordered from dimension 1 to m.

The mixed function essentially compiles response probabilities computed using the functions drm, grm, gpcm, nrm, and mcm for the associated models respectively. All of the arguments specified in any one of these functions can be passed to mixed as an additional argument. For example, the argument incorrect can be passed to drm and catprob can be passed to grm. In the functions drm, grm, and gpcm there is an argument D for the value of a scaling constant. In mixed, a single argument D can be passed that will be applied to all applicable models, or arguments D.drm, D.grm, and D.gpcm can be specified for each model respectively. If an argument is specified for D and,

say `D.drm`, the values for `D.grm` and `D.gpcm` (if applicable) will be set equal to `D`. If only `D.drm` is specified, the values for `D.grm` and `D.gpcm` (if applicable) will be set to 1.

Value

Returns an object of class `irt.prob`

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

See Also

`plot`: plot item characteristic/category curves
`irt.prob`, `irt.pars`, `sep.pars`: classes

Examples

```
###### Unidimensional Examples ######
# Compute probabilities for three dichotomous (3PL) items and two
# polytomous (gpcm) items without a location parameter
dichot <- matrix(c(1.2, .8, .9, 2.3, -1.1, -.2, .24, .19, .13),3,3)
poly <- matrix(c(.64, -1.8, -.73, .45, NA, .88, .06, 1.4, 1.9, 2.6),
  2,5,byrow=TRUE)
pars <- rbind(cbind(dichot,matrix(NA,3,2)),poly)
cat <- c(2,2,2,4,5)
pm <- as.poly.mod(5, c("drm","gpcm"), list(1:3,4:5))
x <- mixed(pars, cat, pm)
plot(x)

# Specify a different scaling constant for the GPCM items in the
# above example
x <- mixed(pars, cat, pm, D.gpcm=1.7)
plot(x)


# Compute probabilities for three dichotomous (3PL) items, four
# polytomous items, two gpcm items and two nrm items. Include a
# location parameter for the gpcm items
a <- matrix(c(
  1.2, rep(NA,4),
  .8, rep(NA,4),
  .9, rep(NA,4),
  .64, rep(NA,4),
  .88, rep(NA,4),
  .905, .522, -.469, -.959, NA,
  .828, .375, -.357, -.079, -.817),7,5,byrow=TRUE)
b <- matrix(c(
  2.3, rep(NA,4),
  -1.1, rep(NA,4),
  -.2, rep(NA,4),
  -.69, -1.11, -.04, 1.14, NA,
```

```
   1.49, -1.43, -.09, .41, 1.11,
   .126, -.206, -.257, .336, NA,
   .565, .865, -1.186, -1.199, .993),7,5,byrow=TRUE)
c <- c(.14, .19, .26, rep(NA,4))
pars <- list(a,b,c)
cat <- c(2,2,2,4,5,4,5)
pm <- as.poly.mod(7, c("drm","gpcm","nrm"), list(1:3,4:5,6:7))
x <- mixed(pars, cat, pm, location=TRUE)
plot(x)

###### Multidimensional Example ######
# Compute response probabilities for four dichotomous items
# modeled using the M2PL and three polytomous items modeled
# using the multidimensional graded response model. For the
# later items, cumulative probabilities are computed.
a <- matrix(c(1.66,1.72,.69,.19,.88,1.12,.68,1.21,
      .873, .226, .516, .380, .613, .286 ),7,2,byrow=TRUE)
d <- matrix(c(-.38,NA,NA,NA,NA,
      -.68,NA,NA,NA,NA,
      -.91,NA,NA,NA,NA,
      -1.08,NA,NA,NA,NA,
      2.255, 1.334, -.503, -2.051, -3.082,
      1.917, 1.074, -.497, -1.521, -2.589,
      1.624, .994, -.656, -1.978, NA),7,5,byrow=TRUE)
cat <- c(2,2,2,2,6,6,5)
pars <- cbind(a,d)
pm <- as.poly.mod(7,c("drm","grm"),list(1:4,5:7))
x <- mixed(pars, cat, pm, dimensions=2, catprob=TRUE)
plot(x)
```

---

nrm-methods   *Nominal Response Model Probabilities*

Description

This function computes the probability of responding in a specific category for one or more items for a given set of theta values using the nominal response model or multidimensional nominal response model.

Usage

```
nrm(x, cat, theta, dimensions = 1, items, information = FALSE, angle, ...)

## S4 method for signature 'matrix', 'numeric'
nrm(x, cat, theta, dimensions, items, information, angle, ...)

## S4 method for signature 'data.frame', 'numeric'
nrm(x, cat, theta, dimensions, items, information, angle, ...)
```

```
## S4 method for signature 'list', 'numeric'
nrm(x, cat, theta, dimensions, items, information, angle, ...)

## S4 method for signature 'irt.pars', 'ANY'
nrm(x, cat, theta, dimensions, items, information, angle, ...)

## S4 method for signature 'sep.pars', 'ANY'
nrm(x, cat, theta, dimensions, items, information, angle, ...)
```

Arguments

x	Object containing item parameters. See below for more details.
cat	vector identifying the number of response categories for each item
theta	vector, matrix, or list of theta values for which probabilities will be computed. If theta is not specified, an equal interval range of values from -4 to 4 is used with an increment of 0.5. See details below for more information.
dimensions	number of modeled dimensions
items	numeric vector identifying the items for which probabilities should be computed
information	logical value. If TRUE compute item information. In the multidimensional case, information will be computed in the directions specified by angle or default angles of 0 - 90 in increments of 10 degrees.
angle	vector or matrix of angles between the dimension 1 axis and the corresponding axes for each of the other dimensions for which information will be computed. When there are more than two dimensions and angle is a vector, the same set of angles will be used relative to each of the corresponding axes.
...	further arguments passed to or from other methods

Details

theta can be specified as a vector, matrix, or list. For the unidimensional case, theta should be a vector. If a matrix or list of values is supplied, they will be converted to a single vector of theta values. For the multidimensional case, if a vector of values is supplied it will be assumed that this same set of values should be used for each dimension. Probabilities will be computed for each combination of theta values. Similarly, if a list is supplied, probabilities will be computed for each combination of theta values. In instances where probabilities are desired for specific combinations of theta values, a j x m matrix should be specified for j ability points and m dimensions where the columns are ordered from dimension 1 to m.

Value

Returns an object of class [irt.prob](#)

Methods

x = "matrix", cat = "numeric" This method allows one to specify an n x (m x 2k) matrix for n items, m dimensions, and k equal to the maximum number of response categories across items. The first (m x k) columns are for category slope parameters and the remaining columns are

for the category difficulty parameters. For any items with fewer categories than the maximum, the remaining cells in each block of (m x k) columns should be NA.

Unidimensional Specification: Say we have one four category item and one five category item, the first four columns of the four response item would include the slope parameters. The fifth column for this item would be NA. The next four columns would include the category difficulty values, and the last column would be NA.

Multidimensional Specification: In the multidimensional case, the columns for the slope and difficulty parameters should be grouped first by dimension and then by category. Using the same example for the two items with two dimensions there will be 20 columns. The first four columns for the four category item would include the slope parameters associated with the first dimension for each of the four categories respectively. Columns 9-10 would be NA. Columns 11-14 would include the category difficulties associated with the first dimension and columns 19-20 would be NA.

x = "data.frame", cat = "numeric" See the method for x = "matrix"

x = "list", cat = "numeric" This method is for a list with two elements. The first element is an n x (m x k) matrix of category slope values for n items, m dimensions, and k equal to the maximum number of response categories across items. The second list element is an n x (m x k) matrix of category difficulty parameters. For either element, for items with fewer categories than the maximum, the remaining cells in the rows should be NA (see the examples for method x = "matrix" for specification details).

x = "irt.pars", cat = "ANY" This method can be used to compute probabilities for the nrm items in an object of class "irt.pars". If x contains dichotomous items or items associated with another polytomous model, a warning will be displayed stating that probabilities will be computed for the nrm items only. If x contains parameters for multiple groups, a list of "irt.prob" objects will be returned.

x = "sep.pars", cat = "ANY" This method can be used to compute probabilities for the mcm items in an object of class sep.pars. If x contains dichotomous items or items associated with another polytomous model, a warning will be displayed stating that probabilities will be computed for the nrm items only.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Bock, R.D. (1972) Estimating item parameters and latent ability when responses are scored in two or more nominal categories. *Psychometrika*, *37*(1), 29-51.

Bock, R.D. (1996) The nominal categories model. In W.J. van der Linden & Hambleton, R. K. (Eds.) *Handbook of Modern Item Response Theory*. New York: Springer-Verlag

Bolt, D. M. & Johnson, T. J. (in press) Applications of a MIRT model to self-report measures: Addressing score bias and DIF due to individual differences in response style. *Applied Psychological Measurement*.

Kolen, M. J., & Brennan, R. L. (2004) *Test Equating, Scaling, and Linking.* New York: Springer

Takane, Y., & De Leeuw, J. (1987) On the relationship between item response theory and factor analysis of discretized variables. *Psychometrika*, *52*(3), 393-408.

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL <http://www.jstatsoft.org/v35/i12/>

See Also

`mixed`: compute probabilities for mixed-format items
`plot`: plot item characteristic/category curves
`irt.prob`, `irt.pars`, `sep.pars`: classes

Examples

```
###### Unidimensional Example ######
## Item parameters from Bock (1972, p. 46,47)
a <- matrix(c(.905, .522, -.469, -.959, NA,
  .828, .375, -.357, -.079, -.817)), 2,5,byrow=TRUE)
c <- matrix(c(.126, -.206, -.257, .336, NA,
  .565, .865, -1.186, -1.199, .993), 2,5,byrow=TRUE)
pars <- cbind(a,c)
x <- nrm(pars, c(4,5))
plot(x,auto.key=list(space="right"))

###### Multidimensional Example ######
# From Bolt & Johnson (in press)
pars <- matrix(c(-1.28, -1.029, -0.537, 0.015, 0.519, 0.969, 1.343,
1.473, -0.585, -0.561, -0.445, -0.741, -0.584, 1.444,
0.29, 0.01, 0.04, 0.34, 0, -0.04, -0.63), 1,21)
x <- nrm(pars, cat=7, dimensions=2)
# Plot separated surfaces
plot(x,separate=TRUE,drape=TRUE)
```

---

plink-methods   *Separate Calibration*

Description

This function conducts separate calibration of unidimensional or multidimensional IRT single-format or mixed-format item parameters for multiple groups. The unidimensional methods include the Mean/Mean, Mean/Sigma, Haebara, and Stocking-Lord methods. The multidimensional methods include a least squares approach (Li & Lissitz, 2000; Min, 2003; Reckase & Martineau, 2004) and extensions of the Haebara and Stocking-Lord method using a single dilation parameter (Li & Lissitz, 2000), multiple dilation parameters (Min, 2003), or an oblique Procrustes approach (Oshima, Davey, & Lee, 2000; Reckase, 2009). The methods allow for symmetric and non-symmetric optimization and chain-linked rescaling of item and ability parameters.

Usage

```
plink(x, common, rescale, ability, method, weights.t, weights.f,
  startvals, exclude, score = 1, base.grp = 1, symmetric = FALSE,
  rescale.com = TRUE, grp.names = NULL, dilation = "oblique",
```

```
    md.center = FALSE, dim.order = NULL, ...)

## S4 method for signature 'list', 'matrix'
plink(x, common, rescale, ability, method, weights.t, weights.f,
  startvals, exclude, score, base.grp, symmetric, rescale.com,
  grp.names, dilation, md.center, dim.order, ...)

## S4 method for signature 'list', 'data.frame'
plink(x, common, rescale, ability, method, weights.t, weights.f,
  startvals, exclude, score, base.grp, symmetric, rescale.com,
  grp.names, dilation, md.center, dim.order, ...)

## S4 method for signature 'list', 'list'
plink(x, common, rescale, ability, method, weights.t, weights.f,
  startvals, exclude, score, base.grp, symmetric, rescale.com,
  grp.names, dilation, md.center, dim.order, ...)

## S4 method for signature 'irt.pars', 'ANY'
plink(x, common, rescale, ability, method, weights.t, weights.f,
  startvals, exclude, score, base.grp, symmetric, rescale.com,
  grp.names, dilation, md.center, dim.order, ...)
```

Arguments

x	an object of class [irt.pars](#) with multiple groups or a list of irt.pars and/or [sep.pars](#) objects.
common	matrix or list of common items. See below for more details.
rescale	if missing (default), the parameters in x will not be transformed to the base scale. To transform the parameters use "MM","MS","HB","SL","LS" for the Mean/Mean, Mean/Sigma, Haebara, Stocking-Lord, and least squares linking constants respectively.
ability	list of theta values with length equal to the number of groups. If supplied, these values will be transformed to the base scale using the constants identified in rescale or the Haebara constants if rescale is missing.
method	character vector identifying the linking methods to use. Values can include "MM","MS","HB","SL","LS" for the Mean/Mean, Mean/Sigma, Haebara, Stocking-Lord, and least squares linking constants respectively, or if missing, constants will be estimated for all methods.
weights.t	list containing information about the theta values and weights on the *To* scale for use with the characteristic curve methods. See below for more details.
weights.f	list containing information about the theta values and weights on the *From* scale for use with the characteristic curve methods. This argument will be ignored if symmetric=FALSE. See below for more details.
startvals	vector or list of slope and intercept starting values for the characteristic curve methods or character value(s) of "MM", "MS", or "LS" to identify that values from the Mean/Mean, Mean/Sigma, or least squares method should be used for the starting values. See below for more details.

exclude	character vector or list identifying common items that should be excluded when estimating the linking constants. See below for more details.
score	if score = 1, score responses for the Stocking-Lord method with zero for the lowest category and k-1 for the highest, k, category for each item. If score = 2, score responses with one for the lowest category and k for the highest, k, category for each item. A vector or list of scoring weights for each response category can be supplied, but this is only recommended for advanced users.
base.grp	integer identifying the group for the base scale
symmetric	if TRUE use symmetric minimization for the characteristic curve methods. See Kim and Lee (2006) for more information
rescale.com	if TRUE rescale the common item parameters using the estimated linking constants; otherwise, insert the non-transformed common item parameters into the set of unique transformed item parameters
grp.names	character vector of group names
dilation	character value identifying whether an oblique Procrustes approach (Oshima, Davey, & Lee, 2000; Reckase 2009) "oblique", an orthogonal Procrustes approach with a fixed dilation parameter (Li & Lissitz, 2000) "orth.fd", or an orthogonal Procrustes approach with variable dilations (Min, 2003) "orth.vd" should be used to estimate the linking constants for the multidimensional methods. Both orthogonal approaches can be constrained to exclude reflections by using "orth.fdc" or "orth.vdc".
md.center	if TRUE mean center the slope matrices prior to estimating the rotation matrix and dilation constants
dim.order	matrix for identifying the ordering of factors across groups. See below for details.
...	further arguments passed to or from other methods. See below for details.

Details

If x contains only two elements, common should be a matrix. If x contains more than two elements, common should be a list. In any of the common matrices the first column identifies the common items for the first group of two adjacent list elements in x. The second column in common identifies the corresponding set of common items from the next list element in x. For example, if x contains only two list elements, a single set of common items links them together. If item 4 in group one (row 4 in slot pars) is the same as item 6 in group two, the first row of common would be (4,6).

startvals can be a vector or list of starting values for the slope(s) and intercept(s). For unidimensional methods, when there are only two groups, this argument should be a vector of length of two with the first value for the slope and the second value for the intercept or a character value equal to "MM" or"MS". When there are more than two groups a vector of starting values or a character value can be supplied (the same numeric values, if a vector is supplied, will be used for all pairs, or the corresponding MM/MS values for each pair of tests will be used) or a list of vectors/character values can be supplied with the number of list elements equal to the number of groups minus one. For the multidimensional methods, the same general structure applies (a vector or character value for a single group or a vector, character value or list for multiple groups); however, the length of the vector will vary depending on the dilation approach used. If dilation is "obliquw", the first m*m values in startvals, for m dimensions, should correspond to the values in the transformation

matrix (starting with the value in the upper-left corner, then the next value in the column, ..., then the first value in the next column, etc.). The remaining m values should be for the translation vector. If `dilation` is "orth.fd", the first value will be the slope parameter and the remaining m values will be for the translation vector. If `dilation` is "orth.vd", the first m values are the slopes for each dimension and the remaining m values are for the translation vector.

`weights.t` can be a list or a list of lists. The purpose of this object is to specify the theta values on the *To* scale to integrate over in the characteristic curve methods as well as any weights associated with the theta values. See Kim and Lee (2006) or Kolen and Brennan (2004) for more information of these weights. The function `as.weight` can be used to facilitate the creation of this object. If `weights.t` is missing, the default–in the unidimensional case–is to use equally spaced theta values ranging from -4 to 4 with an increment of 0.05 and theta weights equal to one for all theta values. In the multidimensional case, the default is to use 1000 randomly sampled values from a multivariate normal distribution with correlations equal to 0.6 for all dimensions. The theta weights are equal to the normal distribution weights at these points.

To better understand the elements of `weights.t`, let us assume for a moment that x has parameters for only two groups and that we are using non-symmetric linking. In this instance, `weights.t` would be a single list with length two. The first element should be a vector of theta values corresponding to points on the *To* scale. The second list element should be a vector of weights corresponding the theta values. If x contains multiple groups, a single `weights.t` object can be supplied, and the same set of thetas and weights will be used for all adjacent groups. However, a separate list of theta values and theta weights for each adjacent group in x can be supplied.

The specification of `weights.f` is the same as that for `weights.t`, although the theta values and weights for this object correspond to theta values on the *From* scale. This argument is only used when `symmetric=TRUE`. If `weights.f` is missing and `symmetric=TRUE`, the same theta values and weights used for `weights.t` will be used for `weights.f`.

In general, all of the common items identified in x or `common` will be used in estimating linking constants; however, there are instances where there is a need to exclude certain common items (e.g., NRM or MCM items or items exhibiting parameter drift). Instead of creating a new matrix or list of matrices for `common`, the `exclude` argument can be used. `exclude` can be specified as a character vector or a list. In the former case, a vector of model names (i.e., "drm", "grm", "gpcm", "nrm", "mcm") would be supplied, indicating that any common item on any test associated with the given model(s) would be excluded from the set of items used to estimate the linking constants. If the argument is specified as a list, `exclude` should have as many elements as groups in x. Each list element can include model names and/or item numbers corresponding to the common items that should be excluded for the given group. If no items need to be excluded for a given group, the list element should be NULL or NA. For example, say we have two groups and we would like to exclude the NRM items and item 23 from the first group, we would specify `exclude` as `exclude <- list(c("nrm",23),NA)`. Notice that the item number corresponding item 23 in group 2 does not need to be specified.

The argument `dim.order` is a k x r matrix for k groups and r unique dimensions across groups. This object identifies the common dimensions across groups. The elements in the matrix should correspond to the dimension (i.e., the column in the matrix of slope parameters) for a given group. For example, say there are four unique dimensions across two groups, each group only measures three dimensions, and there are only two common dimensions. We might specify a matrix as follows `dim.order <- matrix(c(1:3,NA,NA,1:3),2,4)`. In words, this means that dimensions 2 and 3 in the first group correspond to dimensions 1 and 2 in the second group respectively. If no `dim.order` is specified, it is assumed that all of the dimensions are common, or in instances with different

numbers of factors, that the first m dimensions for each group are common and the remaining r-m dimensions for a given group are unique.

For the characteristic curve methods, response probabilities are computed using the functions drm, grm, gpcm, nrm, and mcm for the associated models respectively. Various arguments from these functions can be passed to plink. Specifically, the argument incorrect can be passed to drm and catprob can be passed to grm. In the functions drm, grm, and gpcm there is an argument D for the value of a scaling constant. In plink, a single argument D can be passed that will be applied to all applicable models, or arguments D.drm, D.grm, and D.gpcm can be specified for each model respectively. If an argument is specified for D and, say D.drm, the values for D.grm and D.gpcm (if applicable) will be set equal to D. If only D.drm is specified, the values for D.grm and D.gpcm (if applicable) will be set to 1.

Value

Returns an object of class [link](#). The labels for the linking constants are specified in the following manner "group1/group2", meaning the group1 parameters were transformed to the group2 test. The base group is indicated by an asterisk.

Methods

x = "list", common = "matrix" This method is used when x contains only two list elements. If either of the list elements is of class irt.pars, they can include multiple groups. common is the matrix of common items between the two groups in x. See details for more information on common.

x = "list", common = "data.frame" See the method for signature x="list", common="matrix".

x = "list", common = "list" This method is used when x includes two or more list elements. When x has length two, common (although a single matrix) should be a list with length one. If x has more than two list elements common identifies the common items between adjacent list elements. If objects of class irt.pars are included with multiple groups, common should identify the common items between the first or last group in the irt.pars object, depending on its location in x, and the adjacent list element(s) in x. For example, if x has three elements: an irt.pars object with one group, an irt.pars object with four groups, and a sep.pars object, common will be a list with length two. The first element in common is a matrix identifying the common items between the items in the first irt.pars object and the first group in the second irt.pars object. The second element in common should identify the common items between the fourth group in the second irt.pars object and the items in the sep.pars object.

x = "irt.pars", common = "ANY" This method is intended for an irt.pars object with multiple groups.

Note

The translation vector for the multidimensional Stocking-Lord method may converge to odd values. The least squares method has been shown to produce more accurate constants than the characteristic curve methods (in less time); however, if use of a characteristic curve approach is desired, it is recommended that the Haebara method be used and/or the relative tolerance for the optimization be lowered to 0.001 using the argument control=list(rel.tol=0.001).

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Haebara, T. (1980). Equating logistic ability scales by a weighted least squares method. *Japanese Psychological Research*, *22*(3), 144-149.

Kim, S. & Lee, W.-C. (2006). An Extension of Four IRT Linking Methods for Mixed-Format Tests. *Journal of Educational Measurement*, *43*(1), 53-76.

Kolen, M. J. & Brennan, R. L. (2004) *Test Equating, Scaling, and Linking* (2nd ed.). New York: Springer

Li, Y. H., & Lissitz, R. W. (2000). An evaluation of the accuracy of multidimensional IRT linking. *Applied Psychological Measurement*, *24*(2), 115-138.

Loyd, B. H. & Hoover, H. D. (1980). Vertical Equating Using the Rasch Model. *Journal of Educational Measurement*, *17*(3), 179-193.

Marco, G. L. (1977). Item Characteristic Curve Solutions to Three Intractable Testing Problems. *Journal of Educational Measurement*, *14*(2), 139-160.

Min, K. -S. (2007). Evaluation of linking methods for multidimensional IRT calibrations. *Asia Pacific Education Review*, *8*(1), 41-55.

Oshima, T. C., Davey, T., & Lee, K. (2000). Multidimensional linking: Four practical approaches. *Journal of Educational Measurement*, *37*(4), 357-373.

Reckase, M. D. (2009). *Multidimensional item response theory* New York, Springer

Reckase, M. D., & Martineau, J. A. (2004). *The vertical scaling of science achievement tests*. Research report for the Center for Education and National Research Council.

Stocking, M. L. & Lord, F. M. (1983). Developing a common metric in item response theory. *Applied Psychological Measurement*, *7*(2), 201-210.

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL http://www.jstatsoft.org/v35/i12/

Examples

```
###### Unidimensional Examples ######
# Create irt.pars object with two groups (all dichotomous items),
# rescale the item parameters using the Mean/Sigma linking constants,
# and exclude item 27 from the common item set
pm <- as.poly.mod(36)
x <- as.irt.pars(KB04$pars, KB04$common, cat=list(rep(2,36),rep(2,36)),
  poly.mod=list(pm,pm))
out <- plink(x, rescale="MS", base.grp=2, D=1.7, exclude=list(27,NA))
summary(out, descrip=TRUE)
pars.out <- link.pars(out)


# Create object with six groups (all dichotomous items)
pars <- TK07$pars
common <- TK07$common
```

```
cat <- list(rep(2,26),rep(2,34),rep(2,37),rep(2,40),rep(2,41),rep(2,43))
pm1 <- as.poly.mod(26)
pm2 <- as.poly.mod(34)
pm3 <- as.poly.mod(37)
pm4 <- as.poly.mod(40)
pm5 <- as.poly.mod(41)
pm6 <- as.poly.mod(43)
pm <- list(pm1, pm2, pm3, pm4, pm5, pm6)
x <- as.irt.pars(pars, common, cat, pm,
  grp.names=paste("grade",3:8,sep=""))
out <- plink(x)
summary(out)
constants <- link.con(out) # Extract linking constants

# Create an irt.pars object and a sep.pars object for two groups of
# nominal response model items. Compare non-symmetric and symmetric
# minimization. Note: This example may take a minute or two to run
## Not run:
pm <- as.poly.mod(60, "nrm", 1:60)
pars1 <- as.irt.pars(act.nrm$yr97, cat=rep(5,60), poly.mod=pm)
pars2 <- sep.pars(act.nrm$yr98, cat=rep(5,60), poly.mod=pm)
out <- plink(list(pars1, pars2), matrix(1:60,60,2))
out1 <- plink(list(pars1, pars2), matrix(1:60,60,2), symmetric=TRUE)
summary(out, descrip=TRUE)
summary(out1, descrip=TRUE)

## End(Not run)

# Compute linking constants for two groups with multiple-choice model
# item parameters. Rescale theta values and item parameters using
# the Haebara linking constants
# Note: This example may take a minute or two to run
## Not run:
theta <- rnorm(100) # In practice, estimated theta values would be used
pm <- as.poly.mod(60, "mcm", 1:60)
x <- as.irt.pars(act.mcm, common=matrix(1:60,60,2), cat=list(rep(6,60),
  rep(6,60)), poly.mod=list(pm,pm))
out <- plink(x, ability=list(theta,theta), rescale="HB")
pars.out <- link.pars(out)
ability.out <- link.ability(out)
summary(out, descrip=TRUE)

## End(Not run)

# Compute linking constants for two groups using mixed-format items and
# a mixed placement of common items. Compare calibrations with the
# inclusion or exclusion of NRM items. This example uses the dgn dataset.
pm1 <- as.poly.mod(55,c("drm","gpcm","nrm"),dgn$items$group1)
pm2 <- as.poly.mod(55,c("drm","gpcm","nrm"),dgn$items$group2)
x <- as.irt.pars(dgn$pars,dgn$common,dgn$cat,list(pm1,pm2))
# Run with the NRM common items included
out <- plink(x)
# Run with the NRM common items excluded
```

```
out1 <- plink(x,exclude="nrm")
summary(out)
summary(out1)


# Compute linking constants for six groups using mixed-format items and
# a mixed placement of common items. This example uses the reading dataset.
# See the information on the dataset for an interpretation of the output.
pm1 <- as.poly.mod(41, c("drm", "gpcm"), reading$items[[1]])
pm2 <- as.poly.mod(70, c("drm", "gpcm"), reading$items[[2]])
pm3 <- as.poly.mod(70, c("drm", "gpcm"), reading$items[[3]])
pm4 <- as.poly.mod(70, c("drm", "gpcm"), reading$items[[4]])
pm5 <- as.poly.mod(72, c("drm", "gpcm"), reading$items[[5]])
pm6 <- as.poly.mod(71, c("drm", "gpcm"), reading$items[[6]])
pm <- list(pm1, pm2, pm3, pm4, pm5, pm6)
x <- as.irt.pars(reading$pars, reading$common, reading$cat, pm, base.grp=4)
out <- plink(x)
summary(out)


###### Multidimensional Examples ######
# Reckase Chapter 9
pm <- as.poly.mod(80, "drm", list(1:80))
x <- as.irt.pars(reckase9$pars, reckase9$common,
  list(rep(2,80),rep(2,80)), list(pm,pm), dimensions=2)
# Compute constants using the least squares method and
# the orthongal rotation with variable dilation.
# Rescale the item parameters using the LS method
out <- plink(x, dilation="orth.vd", rescale="LS")
summary(out, descrip=TRUE)
# Extract the rescaled item parameters
pars.out <- link.pars(out)

# Compute constants using an oblique Procrustes method
# Display output and descriptives
out <- plink(x, dilation="oblique")
summary(out, descrip=TRUE)

# Compute constants using and orthogonal rotation with
# a fixed dilation parameter
# Rescale the item parameters and ability estimates
# using the "HB" and "LS" methods.
# Display the optimization iterations
ability <- matrix(rnorm(40),20,2)
out <- plink(x, method=c("HB","LS"), dilation="orth.fd",
   rescale="HB", ability=list(ability,ability),
   control=list(trace=1,rel.tol=0.001))
summary(out)
# Extract rescaled ability estimates
ability.out <- out$ability


# Compute linking constants for two groups using mixed-format items
```

```
# modeled with the M3PL and MGPCM. Only compute constants using the
# orth.vd dilation.
pm <- as.poly.mod(60,c("drm","gpcm"), list(c(1:60)[md.mixed$cat==2],
  c(1:60)[md.mixed$cat>2]))
x <- as.irt.pars(md.mixed$pars, matrix(1:60,60,2),
  list(md.mixed$cat, md.mixed$cat),
  list(pm, pm), dimensions=4, grp.names=c("Form.X","Form.Y"))
out <- plink(x,dilation="orth.vd")
summary(out, descrip=TRUE)


# Illustration of construct shift and how to specify common dimensions
pm <- as.poly.mod(80, "drm", list(1:80))
pars <- cbind(round(runif(80),2),reckase9$pars[[1]])
x <- as.irt.pars(list(pars,reckase9$pars[[2]]), reckase9$common,
list(rep(2,80),rep(2,80)), list(pm,pm), dimensions=c(3,2))
dim.order <- matrix(c(1,2,3,NA,1,2),2,3,byrow=TRUE)
out <- plink(x, dilation="oblique", dim.order=dim.order, rescale="LS")
summary(out)
pars.out <- link.pars(out)
```

---

plot.irt.prob   *Plot Item Stuff*

Description

This function plots item response curves/surfaces using the lattice package in addition to vector plots, contour plots, and level plots for multidimensional items, and comparison plots for examining parameter drift.

Usage

```
## S3 method for signature 'irt.pars'
## S3 method for class 'irt.pars'
plot(x, y, ..., type, separate, combine, items, item.names,
    item.nums, panels, drift, groups, grp.names, sep.mod,
    drift.sd, save.hist)

## S3 method for signature 'irt.prob'
## S3 method for class 'irt.prob'
plot(x, y, ..., type, separate, combine, items, item.names,
    item.nums, panels, save.hist)

## S3 method for signature 'sep.pars'
## S3 method for class 'sep.pars'
plot(x, y, ..., type, separate, combine, items, item.names,
    item.nums, panels, save.hist)
```

```
## S3 method for signature 'list'
## S3 method for class 'irt.pars'
plot(x, y, ..., type, separate, combine, items, item.names,
    item.nums, panels, drift, groups, grp.names, sep.mod,
    drift.sd, save.hist)
```

Arguments

x	object of class `irt.prob`, `irt.pars`, or `sep.pars`. For the later two classes, probabilities are computed using the function `mixed` before plotting the curves.
y	this is an argument in the generic plot function, but is not used for these methods
...	further arguments passed to or from other methods
type	type of plot to produce (for multidimensional models). Values can include "wire-frame", "contourplot", "levelplot", "vectorplot1", "vectorplot2", "vectorplot3". The default if `type` is missing is "wireframe". The first three types of plots are based on the correspondingly named lattice plots; properties can be controlled using arguments associated with the respective methods. See the details below for more information on multidimensional plots.
separate	logical value. If `TRUE`, plot the item category curves or surfaces for polytomous items in separate panels
combine	vector identifying the number of response categories to plot in each panel. If `NULL`, the curves will be grouped by item. `combine` is typically used to plot curves for more than one item in a panel.
items	numeric vector identifying the items to plot
item.names	vector of item names for use in labeling the panels
item.nums	logical value. If `TRUE`, include item numbers on the vector plots.
panels	number of panels to display in the output window. If the number of items is greater than `panels`, the plots will be created on multiple pages. If `NULL`, all panels will be plotted on a single page.
drift	character vector identifying the plots to create to examine item parameter drift. Acceptable values are "a", "b", "c" for the various parameters respectively, "pars" to compare all of these parameters, "TCC" to compare test characteristic curves, "ICC" to compare item characteristic curves, or "all" to produce all of these plots.
groups	numeric vector identifying the groups in x that should be included for comparing parameter drift. The values should correspond to the group number of the lowest group of each pair of adjacent groups in x.
grp.names	character vector of group names to use when creating the drift plots
sep.mod	logical value. If `TRUE` use different markers in the drift plots to identify parameters related to different item response models
drift.sd	numeric value identifying the number of standard deviations to use when creating the perpendicular confidence region for the drift comparison plots. If missing, `drift.sd` will default to 3.
save.hist	logical value. If `FALSE`, the plot history should be deleted (only applies to Windows); otherwise, any previously saved plots will be retained.

Details

All of the plots, with the exception of the vector plots, are based on the `lattice` package. For the unidimensional plots, any arguments associated with the `xyplot` function can be included. For the multidimensional plots, any arguments associated with the `wireframe`, `contourplot`, or `levelplot` functions (corresponding to the argument specified in `type`) can be passed as well.

For the multidimensional plots, vector plots are limited to two dimensions. The other three types of plots can be used with multiple dimensions by plotting the results for the first two dimensions conditional on each combination of theta values for the other dimensions. The function can handle up to ten dimensions; however, the usefulness of the plots will likely diminish if more than four dimensions are modeled.

There are three types of vector plots. The first two types create a set of vectors that originate at the location of the multidimensional difficulty and characterize the length of the vectors at the appropriate angles, relative to the axes, based on the multidimensional discrimination. Setting `type` equal to "vectorplot2" will also include the reference composite based on an eigenvalue decomposition of the slope parameters. The third type of vector plot is similar to the first with the key distinction that the lines are drawn from the origin to the points of multidimensional difficulty. When there are more than two dimensions, a panel of vectorplots is created for each pair of dimensions.

Note

When multiple pages are created, the PgUp and PgDn buttons can be used to view the plots on different pages on a Windows machine. For other operating systems, it is suggested that the multipage plot be sent to an external graphics file.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>, Adam Wyse (Contributing Author)

References

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL http://www.jstatsoft.org/v35/i12/

See Also

irt.prob, irt.pars, sep.pars,

Examples

```
###### Unidimensional Examples ######
# Compute probabilites for three dichotomous (3PL) items and two polytomous
# (gpcm) items then plot the item characteristic/category curves
dichot <- matrix(c(1.2, .8, .9, 2.3, -1.1, -.2, .24, .19, .13),3,3)
poly <- matrix(c(.64, -1.8, -.73, .45, NA, .88, .06, 1.4, 1.9, 2.6),
  2,5,byrow=TRUE)
pars <- rbind(cbind(dichot,matrix(NA,3,2)),poly)
cat <- c(2,2,2,4,5)
pm <- as.poly.mod(5, c("drm","gpcm"), list(1:3,4:5))
```

```
x <- mixed(pars, cat, pm)
plot(x)

# Compute probabilites for six items using the 3PL and plot all of the item
# characteristic curves on a single panel using the combine argument
a <- c(.71,.96,.36,1.05,1.76,.64)
b <- c(-.16,1.18,2.03,1.09,.82,-1.56)
c <- c(.22,.17,.24,.08,.20,.13)
theta <- seq(-3,3,.2)
pars <- cbind(a,b,c)
x <- drm(pars,theta)
plot(x,combine=6,item.names="Items 1-6",auto.key=list(space="right"))

# Compute probabilties for the nominal response model using the ACT
# mathematics data. Plot the item category curves for a subset of 9 items.
x <- nrm(act.nrm[[1]], rep(5,60))
plot(x, items=c(2,7,12,20,35,41,48,57,60),auto.key=list(space="right"))

# Create irt.pars object with two groups (all dichotomous items)
# rescale the item parameters using the Stocking-Lord linking constants
# Create drift plots
pm <- as.poly.mod(36)
x <- as.irt.pars(KB04$pars, KB04$common, cat=list(rep(2,36),rep(2,36)),
  poly.mod=list(pm,pm))
out <- plink(x, rescale="SL", base.grp=2, D=1.7)
plot(out$pars, drift="all", grp.names=c("Form X","Form Y"),
item.nums=TRUE)

## Not run:
# Compute linking constants for six groups using mixed-format items
# Create drift plots with distinct markers for the two response models
pm1 <- as.poly.mod(41, c("drm", "gpcm"), reading$items[[1]])
pm2 <- as.poly.mod(70, c("drm", "gpcm"), reading$items[[2]])
pm3 <- as.poly.mod(70, c("drm", "gpcm"), reading$items[[3]])
pm4 <- as.poly.mod(70, c("drm", "gpcm"), reading$items[[4]])
pm5 <- as.poly.mod(72, c("drm", "gpcm"), reading$items[[5]])
pm6 <- as.poly.mod(71, c("drm", "gpcm"), reading$items[[6]])
pm <- list(pm1, pm2, pm3, pm4, pm5, pm6)
x <- as.irt.pars(reading$pars, reading$common, reading$cat, pm, base.grp=4)
out <- plink(x, rescale="HB")
plot(out$pars, drift=c("a","b"), sep.mod=TRUE)

## End(Not run)

###### Multidimensional Examples ######
# Compute response probabilities for 20 items from the
# reckase9 data. Probabilites are modeled using the M2PL.
# Each of the six plot types will be created.
x <- drm(reckase9[[1]][[1]][31:50,],dimensions=2)

# Wireframe plot
plot(x, drape=TRUE, item.names=paste("Item",31:50))
# Contour Plot
```

```
plot(x, type="contourplot",labels=FALSE,cuts=20)
# Level Plot
plot(x, type="levelplot",cuts=20)

# Use all the items for the vector plots
x <- drm(reckase9[[1]][[1]],dimensions=2)
# Vector Plot 1
plot(x, type="vectorplot1",item.nums=FALSE)
# Vector Plot 2
plot(x, type="vectorplot2",item.nums=FALSE)
# Vector Plot 3
plot(x, type="vectorplot3",xlim=c(-1.5,1.5),ylim=c(-1.5,1.5))


# Compute response probabilities for a single three-category item using
# the multidimensional generalized partial credit model for three
# dimensions. Plot the conditional item category surfaces in a single
# panel and then the separated item category surfaces in separate panels
pars <- matrix(c(1.1999,0.5997,0.8087,2.1730,-1.4576),1,5)
x <- gpcm(pars,3,dimensions=3,theta=-4:4)

# plot combined item category surfaces. Rotate the plots using the
# screen argument
plot(x, screen=list(z=-30,x=-60))

## Not run:
# plot separated item category surfaces
x <- gpcm(pars,3,dimensions=3)
plot(x,separate=TRUE,drape=TRUE,panels=1)

## End(Not run)
```

---

poly.mod-class   *Class "poly.mod"*

Description

The formal S4 class for poly.mod. This class characterizes the models and associated items for a set of item parameters.

Details

The IRT models associated with the codes:

drm: dichotomous response models (includes 1PL, 2PL, and 3PL)

gpcm: generalized partial credit model (includes the partial credit model)

grm: graded response model

mcm: multiple-choice model

nrm: nominal response model

Objects from the Class

Objects can be created by calls of the form new("poly.mod", ...), but this is not encouraged. Use the function as.poly.mod instead.

Slots

model: character vector identifying all the models associated with the corresponding set of item parameters. The only acceptable models are drm, gpcm, grm, mcm, and nrm. See below for more details.

items: list with the same length as model, where each element identifies the items(rows) in the corresponding set of item parameters associated with the model(s) identified in model.

Note

The names of the list elements for items must correspond to the elements in model. For example, if the poly.mod object is pm and the first element in pm@model is drm,one should be able to reference the associated items by pm@items$drm. If the list elements are unnamed, some functions may not work properly.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

See Also

as.poly.mod, irt.pars, sep.pars, irt.prob

---

read.bilog   *Import Parameters from IRT Software*

Description

This function imports item and/or ability parameters from BILOG-MG 3, PARSCALE 4, MULTI-LOG 7, TESTFACT 4, ICL, BMIRT, and ltm.

Usage

```
read.bilog(file, ability = FALSE, pars.only = TRUE, as.irt.pars = TRUE)

read.parscale(file, ability = FALSE, loc.out = FALSE, pars.only = TRUE,
  as.irt.pars = TRUE)

read.multilog(file, cat, poly.mod, ability = FALSE, contrast = "dev",
  drm.3PL = TRUE, loc.out = FALSE, as.irt.pars = TRUE)

read.testfact(file, ability = FALSE, guessing = FALSE, bifactor = FALSE,
  as.irt.pars = TRUE)
```

```
read.icl(file, poly.mod, ability = FALSE, loc.out = FALSE,
  as.irt.pars = TRUE)

read.bmirt(file, ability = FALSE, sign.adjust = TRUE, loc.out = FALSE,
  pars.only = TRUE, as.irt.pars = TRUE)

read.erm(x, loc.out = FALSE, as.irt.pars = TRUE)

read.ltm(x, loc.out = FALSE, as.irt.pars = TRUE)
```

Arguments

file	filename of file containing the item or ability parameters
ability	if TRUE, `file` contains ability parameters
pars.only	if TRUE, only the item/ability parameters will be imported (i.e., any other information like standard errors will be dropped).
loc.out	if TRUE, the step/threshold parameters will be reformated to be deviations from a location parameter
as.irt.pars	if TRUE, the parameters will be output as an [irt.pars](#) object (this is only applicable to item parameters)
cat	vector with the number of response categories for each item. For multiple-choice model items, `cat` is the number of response categories plus one (the additional category is for 'do not know')
poly.mod	a [poly.mod](#) object. See the documentation for the function [as.irt.pars](#) for more information on creating this object.
contrast	an object identifying the type of contrast(s) used to estimate the various parameters for each item. See below for more details.
drm.3PL	logical value indicating whether the dichotomous items (if applicable) were modeled using the three parameter logistic model (3PL)
guessing	logical value indicating whether a guessing parameter was modeled
bifactor	logical value indicating whether the bifactor model was used to estimate the item/ability parameters
sign.adjust	logical value indicating whether the difficulty/step parameters should be multiplied by -1 to make them consistent with common formulations of multidimensional response models
x	output object from one of the following functions in the eRm package: LLTM, LPCM, LRSM,PCM, RM, or RSM or one of the following functions in the `ltm` package: rasch, ltm, tpm, grm, or gpcm

Details

The file extensions for the item parameter and ability files respectively are as follows: .par and .sco for BILOG-MG, PARSCALE, and MULTILOG, .par and .fsc for TESTFACT, and .par and .ss for BMIRT. For ICL, the file extensions are specified by the user, and for ltm, the name of the output object is specified by the user.

When item parameters are estimated in MULTILOG for models other than the 1PL, 2PL, and GRM, the program estimates (and returns) contrast parameters. MULTILOG implements three types of contrasts: deviation, polynomial, and triangle (see Thissen & Steinberg, 1986 for more information). A single type of contrast can be used for all parameters (a, b, and c) for all items or different contrasts can be specified for individual parameters and individual items. If a single type of contrast is used for all parameters for all items, a character value can be specified for the `contrast` argument: "dev", "poly", or "tri" for the three types of contrasts respectively. When different contrasts are used, `contrast` should be a list of length nine. The list elements should be ordered as follows "dev.a","poly.a", "tri.a","dev.c","poly.c","tri.c","dev.d", "poly.d","tri.d" where the first three elements correspond to the various contrasts for the slope parameters, the next three elements correspond to the contrasts for the category parameters, and the last three elements correspond to the contrasts for the lower asymptote (guessing parameters). There are two approaches that can be implemented using this list 1) character vectors with the model names "drm", "grm", "gpcm", "nrm", and "mcm" indicating that the given parameters for all items associated with the given model should be transformed using the specified contrast. In instances where a model is not included for a given parameter (for any of the contrasts) the parameters will be transformed using deviation contrasts. 2) numeric vectors identifying the contrasts used for given parameters for given items can be specified. It is only necessary to include item numbers for the various parameter/contrast combinations when deviation contrasts are not used. See below for examples of how to formulate this argument.

Value

Returns a data.frame or an object of class irt.pars if as.irt.pars = TRUE.

Note

These functions are currently unable to handle output generated when subtests are used.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Hanson, B. A. (2002). IRT command language [Computer Program]. URL http://www.b-a-h.com/software/irt/icl/

Mair, P & Hatzinger, R. (2007). Extended Rasch modeling: The eRm package for the application of IRT models in R. URL http://www.jstatsoft.org/v20/i09

Muraki, E. & Bock, R. D. (2003). PARSCALE 4: IRT item analysis and test scoring for rating scale data [Computer Program]. Chicago, IL: Scientific Software International. URL http://www.ssicentral.com

Rizopoulos, D. (2006). ltm: An R package for latent variable modeling and item response theory analyses. *Journal of Statistical Software*, *17*(5), 1-25. URL http://www.jstatsoft.org/v17/i05/

Thissen, D. (2003). MULTILOG 7: Multiple, categorical item analysis and test scoring using item response theory [Computer Program]. Chicago, IL: Scientific Software International. URL http://www.ssicentral.com

Thissen, D. & Steinberg, L. (1986). A taxonomy of item response models. *Psychometrika*, *51*(4), 567-577.

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL http://www.jstatsoft.org/v35/i12/

Wood, R., Wilson, D. T., Muraki, E., Schilling, S. G., Gibbons, R., & Bock, R. D. (2003). TEST-FACT 4: Test scoring, item statistics, and item factor analysis [Computer Program]. Chicago, IL: Scientific Software International. URL http://www.ssicentral.com

Yao, L. (2008). BMIRT: Bayesian multivariate item response theory [Computer Program]. Monterey, CA: CTB/McGraw-Hill.

Zimowski, M. F., Muraki, E., Mislevy, R. J., & Bock, R. D. (2003). BILOG-MG 3: Multiple-group IRT analysis and test maintenance for binary items [Computer Program]. Chicago, IL: Scientific Software International. URL http://www.ssicentral.com

Examples

```
# Illustration of how to formulate the contrast argument. Say that we
# have 20 items where the first 15 are modeled using the 3PL and the
# last five are modeled using the GPCM.  For the 3PL items, deviation
# contrasts are commonly used for all of the parameters, whereas with
# the GPCM items, polynomial contrasts are typically used for the slope
# parameters and triangle contrasts are used for the category parameters.
# The contrast argument could be specified as follows

contrast <- vector("list",9)
# Note: the list elements do not need to be named for read.multilog
names(contrast) <- c("dev.a","poly.a","tri.a","dev.c","poly.c","tri.c",
  "dev.d", "poly.d","tri.d")
contrast$poly.a <- 16:20
contrast$tri.c <- 16:20

# The object could alternatively be formatted as follows
contrast <- vector("list",9)
names(contrast) <- c("dev.a","poly.a","tri.a","dev.c","poly.c","tri.c",
  "dev.d","poly.d","tri.d")
contrast$dev.a <- 1:15
contrast$poly.a <- 16:20
contrast$dev.c <- 1:15
contrast$tri.c <- 16:20
contrast$dev.d <- 1:15
```

---

reading　　　　　　　　　　*Multi-Group Mixed-Format Reading Assessment*

Description

This (unidimensional) dataset includes item parameters from a large-scale reading assessment. The parameters were estimated using a combination of the three parameter logistic model (3PL) and the generalized partial credit model (GPCM). There are six sets of parameters which are based on tests

administered in four grades over three years. In particular, the data include a grade 3 and grade 4 test in year 0, a grade 4 and grade 5 test in year 1, and a grade 5 and grade 6 test in year 2. The label for the information related to the grade 3 test is grade3.0 where the value after the decimal indicates the year. Similar labels are used for the other grade/year combinations. This dataset is used for illustrative purposes to show in a multi-group scenario how items from different response models can be mixed together and common items can be in different positions across groups.

Usage

```
reading
```

Format

A list of length four. The first element is a list of length six with item parameter estimates for each grade/year. There is no location parameter for the GPCM items. The second list element is a list identifying the number of response categories for the six grade/year combinations. The third element specifies which items correspond to the different item response models for each grade/year respectively. The last element is a list of common item matrices for each adjacent grade/year group.

Source

Briggs, D. C. & Weeks, J. P. (In Press) The Impact of Vertical Scaling Decisions on Growth Interpretations. *Educational Measurement: Issues and Practices*.

| reckase9 | *Reckase MIRT Multidimensional Item Parameters* |

Description

This dataset includes multidimensional two parameter logistic model (M2PL) item parameter estimates for two dimensions and two groups. The Form 1 items are from a "new" form and the Form 2 items are from an "old" form. There are 16 common items between the two forms.

Usage

```
reckase9
```

Format

A list of length two. The first element pars is a list of length two. Each of the list elements in pars is a 80 x 3 matrix of item parameters. Element one is for Form 1, and element two is for Form 2. The first two columns in each matrix are for the slope parameters and the third column is for the parameter related to item difficulty.

The second list element in reckase9 is a matrix identifying the common items between the two sets of parameters in pars. That is, the first column in common identifies the rows in pars$form1 that are common items. The second column identifies the corresponding set of common items in pars$form2.

Source

Reckase, M. D. (2009). Multidimensional item response theory. New York: Springer.

---

sep.pars-class   *Class "sep.pars"*

Description

The formal S4 class for sep.pars. This class stores a set of separated item parameters and characteristics of these parameters.

Details

a will be an n x m matrix for n items and m dimensions if there are no nominal response model or multiple-choice model items. Otherwise, if nrm or mcm items are included, a will be an n x (k x m) matrix with k equal to the maximum number of response categories across items. If nrm or mcm items are included, the discrimination/slope parameters for the dichotomous response models, the graded response model, partial credit model and generalized partial credit model (and the multidimensional extensions of these models) are listed in the first m columns with all other columns filled with NAs.

b is an n x (k x m) matrix of difficulty, threshold, step, or category parameters (depending on the corresponding model) for n items and m dimensions with k equal to the maximum number of b parameters across all items. For items with b less than k x m, the row is right-filled with NAs.

c will be an n x 1 matrix for n items if there are no multiple-choice model items. Otherwise, if mcm items are included, c will be an n x k matrix with k equal to the maximum number of response categories across items minus one. If mcm items are included, the lower asymptote parameters for the dichotomous response models, are listed in the first column with all other columns filled with NAs. The c values for the 1PL, 2PL, M1PL, and M2PL equal zero.

Objects from the Class

Objects can be created by calls of the form new("sep.pars", ...), but this is not encouraged. Use the function sep.pars instead.

Slots

a: matrix of discrimination/slope parameters

b: matrix of difficulty, threshold, step, and category parameters (depending on the associated IRT model)

c: matrix of lower asymptote parameters and category proportions for the dichotomous response models and multiple-choice model respectively. c is equal to NA for each item for all other models.

cat: vector identifying the number of categories associated with each item. All dichotomous items will have cat values equal to 2. Graded response model and partial credit/generalized partial credit model items will have cat values equal to the number of step/threshold parameters plus one. Nominal response model items will have cat values equal to the number of categories, and multiple-choice model items will have cat values equal to the number of categories plus one (the 'do not know' category).

n: vector identifying the total number of items, the total number of dichotomous and polytomous items, and the number of items associated with each polytomous model.

mod.lab: character vector of labels for the model(s).

location: logical value. If TRUE, the step and/or threshold parameters in slot b for the graded response model and generalized partial credit model include a location parameter.

dimensions: numeric value identifying the number of modeled dimensions.

model: character vector identifying all the models associated with the corresponding set of item parameters. The only acceptable models are drm, gpcm, grm, mcm, and nrm (see class poly.mod for more information).

items: list with the same length as model, where each element identifies the items(rows) in the corresponding set of item parameters associated with the model(s) identified in model.

Extends

Class poly.mod, directly.
Class list.poly, by class poly.mod, distance 2.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

See Also

sep.pars, irt.pars

---

sep.pars-methods   *Separate Item Parameters*

Description

This function splits the item parameters in the specified object into discrimination/slope parameters, difficulty/step/threshold/category parameters, and lower asymptote/category probability parameters.

Usage

```
sep.pars(x, cat, poly.mod, dimensions = 1, location = FALSE,
  loc.out = FALSE, ...)

## S4 method for signature 'numeric'
sep.pars(x, cat, poly.mod, dimensions, location, loc.out, ...)

## S4 method for signature 'matrix'
sep.pars(x, cat, poly.mod, dimensions, location, loc.out, ...)

## S4 method for signature 'data.frame'
sep.pars(x, cat, poly.mod, dimensions, location, loc.out, ...)

## S4 method for signature 'irt.pars'
sep.pars(x, cat, poly.mod, dimensions, location, loc.out, ...)

## S4 method for signature 'list'
sep.pars(x, cat, poly.mod, dimensions, location, loc.out, ...)
```

Arguments

x	Object containing item parameters. For details on the formatting of parameters for specific item response models see the corresponding methods (i.e., drm, gpcm, grm, mcm, and nrm). See the Methods section for as.irt.pars for details on how to format the item parameters when combining parameters from multiple models.
cat	vector identifying the number of response categories for each item. If multiple-choice model items are included, cat for these items should equal the number of response categories plus one (the additional category is for 'do not know')
poly.mod	object of class poly.mod identifying the items associated with each IRT model
dimensions	number of modeled dimensions
location	if TRUE, the step parameters are deviations from a location parameter
loc.out	if TRUE, the step/threshold parameters will be reformated to be deviations from a location parameter
...	further arguments passed to or from other methods

Value

Returns an object of class sep.pars

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

Examples

```
###### Unidimensional Examples ######
# Create object for three dichotomous (1PL) items with difficulties -1, 0, 1
x <- sep.pars(c(-1,0,1))


# Create object for three dichotomous (3PL) items and two polytomous
# (gpcm) items without a location parameter (the parameters are
# formatted as a matrix)
dichot <- matrix(c(1.2, .8, .9, 2.3, -1.1, -.2, .24, .19, .13),3,3)
poly <- matrix(c(.64, -1.8, -.73, .45, NA, .88, .06, 1.4, 1.9, 2.6),
  2,5,byrow=TRUE)
pars <- rbind(cbind(dichot,matrix(NA,3,2)),poly)
cat <- c(2,2,2,4,5)
pm <- as.poly.mod(5, c("drm","gpcm"), list(1:3,4:5))
x <- sep.pars(pars, cat, pm)
summary(x)


# Create object for three dichotomous (3PL) items and two polytomous
# (gpcm) items without a location parameter (the parameters are
# included in a list)
a <- c(1.2, .8, .9, .64, .88)
b <- matrix(c(
  2.3, rep(NA,3),
  -1.1, rep(NA,3),
  -.2, rep(NA,3),
  -1.8, -.73, .45, NA,
  .06, 1.4, 1.9, 2.6),5,4,byrow=TRUE)
c <- c(1.4, 1.9, 2.6, NA, NA)
pars <- list(a,b,c)
cat <- c(2,2,2,4,5)
pm <- as.poly.mod(5, c("drm","gpcm"), list(1:3,4:5))
x <- sep.pars(pars, cat, pm)
summary(x)


# Create object for three dichotomous (3PL) items, four polytomous
# items, two gpcm items and two nrm items. Include a location parameter
# for the gpcm items. Maintain the location parameter in the output.
a <- matrix(c(
  1.2, rep(NA,4),
  .8, rep(NA,4),
  .9, rep(NA,4),
  .64, rep(NA,4),
  .88, rep(NA,4),
  .905, .522, -.469, -.959, NA,
  .828, .375, -.357, -.079, -.817),7,5,byrow=TRUE)
b <- matrix(c(
  2.3, rep(NA,4),
  -1.1, rep(NA,4),
  -.2, rep(NA,4),
```

```
  -.69, -1.11, -.04, 1.14, NA,
  1.49, -1.43, -.09, .41, 1.11,
  .126, -.206, -.257, .336, NA,
  .565, .865, -1.186, -1.199, .993),7,5,byrow=TRUE)
c <- c(.14, .19, .26, rep(NA,4))
pars <- list(a,b,c)
cat <- c(2,2,2,4,5,4,5)
pm <- as.poly.mod(7, c("drm","gpcm","nrm"), list(1:3,4:5,6:7))
x <- sep.pars(pars, cat, pm, location=TRUE, loc.out=TRUE)
summary(x, descrip=TRUE)


# Create irt.pars object with two groups then run sep.pars
pm <- as.poly.mod(36)
x <- as.irt.pars(KB04$pars, KB04$common, cat=list(rep(2,36),rep(2,36)),
  list(pm,pm), grp.names=c("form.x","form.y"))
out <- sep.pars(x)
summary(out, descrip=TRUE)


###### Multidimensional Examples ######
# Create object for three dichotomous (M1PL) items for two dimensions
# with parameters related to item difficulties of -1, 0, 1
x <- sep.pars(c(-1,0,1), dimensions=2)

# Create object for three dichotomous (M3PL) items and two polytomous
# (MGPCM) items without a location parameter for four dimensions
# (the parameters are included in a list)
a <- matrix(c(0.5038, 2.1910, 1.1317, 0.2493,
  2.9831, 0.4811, 0.3566, 0.4306,
  0.2397, 0.2663, 1.5588, 0.5295,
  0.2020, 0.2410, 1.2061, 0.5552,
  0.2054, 0.6302, 0.3152, 0.2037),5,4,byrow=TRUE)
b <- matrix(c(0.5240, rep(NA,3),
  -1.8841, rep(NA,3),
  0.2570, rep(NA,3),
  -1.4207, 0.3041, -0.5450, NA,
  -2.1720, 0.0954, 0.6531, 0.9114),5,4,byrow=TRUE)
c <- c(0.1022, 0.3528, 0.2498, NA, NA)
pars <- list(a,b,c)
cat <- c(2,2,2,4,5)
pm <- as.poly.mod(5, c("drm","gpcm"), list(1:3,4:5))
x <- sep.pars(pars, cat, pm, dimensions=4)
summary(x, descrip=TRUE)
```

---

summary.irt.pars   *Summarize Item Parameters/plink Output*

Description

This function summarizes the item parameters for a given object. If the object is of class [link](#), the function prints the linking constants

Usage

```
## S3 method for class 'irt.pars'
summary(object, ..., descrip = FALSE)

## S3 method for class 'sep.pars'
summary(object, ..., descrip = FALSE)

## S3 method for class 'list'
summary(object, ..., descrip = FALSE)

## S3 method for class 'link'
summary(object, ..., descrip = FALSE)
```

Arguments

object	an R object. See details below
...	further arguments passed to or from other methods
descrip	if TRUE,print descriptive statistics for the item parameters in object.

Details

The method for objects of class irt.pars summarizes the parameters for a single group or multiple groups. If multiple groups are included in the object, parameters will be summarized for each group separately.

The method for objects of class sep.pars summarizes the parameters for a single sep.pars object.

Objects of class list can include any combination of irt.pars or sep.pars objects. The parameters will be summarized for each group separately.

The method for objects of class link prints the linking constants for n-1 groups. If descrip = TRUE, the method summarizes the parameters for all groups separately. The labels are specified in the following manner "group1/group2", meaning the group1 parameters were transformed to the group2 test using the associated constants. The base group is indicated by an asterisk.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>

References

Weeks, J. P. (2010) **plink**: An R package for linking mixed-format tests using IRT-based methods. *Journal of Statistical Software*, **35(12)**, 1–33. URL <http://www.jstatsoft.org/v35/i12/>

See Also

as.irt.pars, irt.pars, sep.pars, sep.pars, link

TK07   *Tong - Kolen (2007)*

Description

This (unidimensional) dataset includes three parameter logistic model (3PL) item parameter estimates from simulated dataset based on items from the 1992 Iowa Test of Basic Skills vocabulary test. The data are for six groups (grades 3-8) with varying numbers of common items between adjacent grades.

Usage

TK07

Format

A list of length two. The first element pars is a list of length six. Each of the list elements in pars is a n x 3 matrix of item parameters where n differs for each grade. The second list element in TK07 is a list of length five identifying the common items between adjacent groups in pars. Each list element in common is a j x 2 matrix for j common items. For example, there are 13 common items between grades 3 and 4. The first list element in common is for the common items between these grades. The first column in the matrix show that items 14-26 in grade 3 are the same as items 1-13 in grade 4. Similar matrices are identified for all other grade pairs.

Source

The parameters were obtained from Dr. Ye Tong at Pearson Educational Measurement and were used in the following article:

Tong, Y. & Kolen, M. J. (2007) Comparisons of methodologies and results in vertical scaling for educational achievement tests. *Applied Measurement in Education*, *20*(2), 227-253.

Index