Package ‘irtoys’

December 15, 2017

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>A Collection of Functions Related to Item Response Theory (IRT)</td>
</tr>
<tr>
<td>Version</td>
<td>0.2.1</td>
</tr>
<tr>
<td>Date</td>
<td>2017-12-15</td>
</tr>
<tr>
<td>Maintainer</td>
<td>Ivailo Partchev</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:partchev@gmail.com">partchev@gmail.com</a></td>
</tr>
<tr>
<td>Description</td>
<td>A collection of functions useful in learning and practicing IRT, which can be combined into larger programs. Provides basic CTT analysis, a simple common interface to the estimation of item parameters in IRT models for binary responses with three different programs (ICL, BILOG-MG, and ltm), ability estimation (MLE, BME, EAP, WLE, plausible values), item and person fit statistics, scaling methods (MM, MS, Stocking-Lord, and the complete Hebaera method), and a rich array of parametric and non-parametric (kernel) plots. Estimates and plots Haberman's interaction model when all items are dichotomously scored.</td>
</tr>
<tr>
<td>LazyLoad</td>
<td>yes</td>
</tr>
<tr>
<td>LazyData</td>
<td>yes</td>
</tr>
<tr>
<td>License</td>
<td>GPL (&gt;= 2)</td>
</tr>
<tr>
<td>Depends</td>
<td>sm, ltm</td>
</tr>
<tr>
<td>Suggests</td>
<td>MASS, knitr</td>
</tr>
<tr>
<td>Collate</td>
<td>'ability.R' 'estimate.R' 'itemfit.R' 'nonpar.R' 'readin.R' 'scale.R' 'tracelines.R' 'utils.R' 'irtoys-package.R' 'interaction.R'</td>
</tr>
<tr>
<td>VignetteBuilder</td>
<td>knitr</td>
</tr>
<tr>
<td>RoxygenNote</td>
<td>6.0.1</td>
</tr>
<tr>
<td>NeedsCompilation</td>
<td>yes</td>
</tr>
<tr>
<td>Author</td>
<td>Ivailo Partchev [aut, cre], Gunter Maris [aut], Tamaki Hattori [ctb]</td>
</tr>
<tr>
<td>Repository</td>
<td>CRAN</td>
</tr>
<tr>
<td>Date/Publication</td>
<td>2017-12-15 13:33:18 UTC</td>
</tr>
</tbody>
</table>
R topics documented:

irtoys-package ................................................. 3
ability ............................................................. 4
api ................................................................. 5
b1 ................................................................. 6
b2 ................................................................. 6
b3 ................................................................. 6
dpv ............................................................... 7
eap ................................................................. 8
erf ................................................................. 9
est ................................................................. 9
iif ................................................................. 12
interactionModel .................................................. 13
irf ................................................................. 14
irfPlot ............................................................ 15
itf ................................................................. 16
mlebme ............................................................ 18
normal.qu .......................................................... 19
npp ................................................................. 20
plot.iif ........................................................... 21
plot.imp ........................................................... 22
plot.irf ........................................................... 23
plot.tif ........................................................... 24
plot.trf ........................................................... 25
qrs ................................................................. 26
read.ip.bilog ...................................................... 27
read.ip.icl ........................................................ 27
read.qu.icl ........................................................ 28
read.resp .......................................................... 29
rim ................................................................. 30
sca ................................................................. 31
sco ................................................................. 32
Scored ............................................................. 33
Scored2pl .......................................................... 33
scoreMetric ....................................................... 34
scp ................................................................. 34
sim ................................................................. 35
tet ................................................................. 36
tgf ................................................................. 36
tia ................................................................. 37
tif ................................................................. 38
trf ................................................................. 39
tsc ................................................................. 40
Unscored ............................................................ 41
wle ................................................................. 41

Index 43
The `irtoys` package contains a bunch of functions potentially useful to those teaching or learning Item Response Theory (IRT). R being particularly good at graphics, there is a rich array of plotting functions to visualize IRT models or assess their fit. Ability estimates can be estimated by MLE, BME, EAP, WLE. Various IRT scaling methods are supported: MM, MS, Stocking-Lord, and Hebaera. Last not least, `irtoys` may be the only package to estimate Haberman’s interaction model, although a new and more powerful package is in the making.

Currently, there are several R packages that can estimate the item parameters in various uni- and multidimensional IRT models, but only several years ago one had to use stand-alone programs that had wildly different and often unwieldy user interfaces. Besides, no single program does everything one needs. One of the original purposes of `irtoys` was to provide a simple and unified interface to some of the most basic functions in ICL, BILOG, and R’s own `ltm`, such that beginners could concentrate on learning IRT rather than syntaxes. Once that these steps have been made, those wishing to take advantage of the full functionality of ICL, BILOG & Co. must still master the syntax of their program of choice.

To take full advantage of `irtoys`, some IRT software is needed. Package `ltm` is automatically loaded. ICL by Brad Hanson can be downloaded from his site, www.b-a-h.com: executables are provided for Windows, Linux, and Macintosh. Because of the technical problems occasionally encountered with this site, I have set up an alternative source:

For Linux: https://www.dropbox.com/s/ks72ow7aznmasqd/icl_linux.tar.gz?dl=0
For Windows: https://www.dropbox.com/s/783ka4q3ifkh4ir/icl_win.zip?dl=0
For Macintosh: https://www.dropbox.com/s/z9kji73r59m4pbf/icl_mac.sea.bin?dl=0

BILOG is commercial software sold by SSI — see www.ssicentral.com for further detail.

On Windows, make sure that the executable files (icl.exe for ICL, BLM1.EXE, BLM2.EXE, and BLM3.EXE for BILOG) are located in a directory that is included in the PATH variable. On Linux, BILOG, being a Windows program, is run with wine, and should also be on a path where wine can
find it. On my machine, I have simply put the three files in `~/wine/drive_c/windows/`. It seems that new versions of wine expect them to be explicitly tagged as executable. On Macintosh, at least `ltm` should work in all cases.

NOTE: Starting with version 0.2.0, function `est` returns a list of three elements: `est` contains the parameter estimates and is thus identical to the output in earlier versions, `se` contains the standard errors, in a similar format, and `vcm` contains the variance-covariance matrices (NULL when using ICL). When passing item parameters to another function that only needs the estimates, `irf(ip)` and `irf(ip$est)` can be used interchangeably. This facilitates using simulated item parameters. A function that does require the complete object is .

Also, function `itf` now returns item fit statistics as a vector rather than a list. Finally, since most of the functions in `irtoys` have been written with the "logistic" metric in mind (i.e., $a_j(\theta_i - b_j)$ rather than $1.7a_j^*(\theta_i - b_j)$), function `est` now estimates item parameters only in the logistic metric.

Author(s)

Ivailo Partchev <partchev@gmail.com>

References


ability Estimate ability

Description

A wrapper around all functions for ability estimation

Usage

ability(resp, ip, method = "WLE", mu = 0, sigma = 1, n = 5)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>resp</code></td>
<td>A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data</td>
</tr>
<tr>
<td><code>ip</code></td>
<td>Item parameters: the object returned by <code>est</code>.</td>
</tr>
<tr>
<td><code>method</code></td>
<td>One of: &quot;MLE&quot;, &quot;BME&quot;, &quot;WLE&quot;, &quot;EAP&quot;, &quot;PV&quot;, &quot;QRS&quot;, &quot;SUM&quot;</td>
</tr>
<tr>
<td><code>mu</code></td>
<td>Mean of the apriori distribution when applicable</td>
</tr>
<tr>
<td><code>sigma</code></td>
<td>Standard deviation of the apriori distribution when applicable method=&quot;ML&quot;. Default is 1.</td>
</tr>
<tr>
<td><code>n</code></td>
<td>The number of plausible values to draw for each person (default is 5).</td>
</tr>
</tbody>
</table>
api

Value

Depends on the method.

Author(s)

Ivailo Partchev

See Also

mlebm, eap, wle, dpv, qrs

Examples

theta <- ability(resp=Scored, method="WLE", ip=Scored2pl)

api

The api appropriateness index

Description

Computes the api appropriateness index, a measure of person fit in IRT models

Usage

api(resp, ip, theta)

Arguments

resp A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data
ip Item parameters: the object returned by est, or its first part.
theta A measure of ability, typically produced with mlebm, wle etc. If missing, ML estimates will be computed automatically.

Value

A vector of length equal to the number of rows in resp, containing the appropriateness indices

Author(s)

Ivailo Partchev
References


Examples

api(Scored, Scored2pl)

b1  

Example item parameters 1PL (Bilog)

Description

Item parameter estimates for the 1PL model, estimated with bilog from the example data set Scored. These are provided because not users will have BILOG-MG available.

Format

The object returned by est

b2  

Example item parameters 2PL (Bilog)

Description

Item parameter estimates for the 2PL model, estimated with bilog from the example data set Scored. These are provided because not users will have BILOG-MG available.

Format

The object returned by est

b3  

Example item parameters 3PL (Bilog)

Description

Item parameter estimates for the 3PL model, estimated with bilog from the example data set Scored. These are provided because not users will have BILOG-MG available.

Format

The object returned by est
**dpv**

- **Draw plausible values**

**Description**

Draws (by rejection sampling) plausible values from the posterior distribution of ability

**Usage**

```r
dpv(resp, ip, mu = 0, sigma = 1, n = 5)
```

**Arguments**

- `resp` A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data
- `ip` Item parameters: the object returned by `est`.
- `mu` Mean of the apriori distribution. Default is 0.
- `sigma` Standard deviation of the apriori distribution. Ignored when `method="ML"`. Default is 1.
- `n` The number of plausible values to draw for each person (default is 5).

**Value**

A matrix with `n` columns

**Author(s)**

Ivailo Partchev

**See Also**

- `mlebme`, `eap`

**Examples**

```r
plval <- dpv(resp=Scored, ip=Scored2pl)
```
**eap**

*EAP estimation of ability*

**Description**

Estimates the expectation of the posterior distribution of the latent variable ("ability") for each person.

**Usage**

```r
eap(resp, ip, qu)
```

**Arguments**

- `resp` A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data
- `ip` Item parameters: the object returned by `est`.
- `qu` A quadrature object produced with `normal.qu` or read in with `read.qu.icl`

**Value**

A matrix with the ability estimates in column 1, and their standard errors of measurement (SEM) in column 2, and the number of non-missing responses in column 3

**Author(s)**

Ivailo Partchev

**See Also**

`mlebme`, `normal.qu`, `read.qu.icl`

**Examples**

```r
th.ep <- eap(resp=Scored, ip=Scored2pl, qu=normal.qu())
```
erf

Plot empirical response function

Description

Produces a plot of the empirical response function (proportion of correct responses at each possible sum score).

Usage

erf(resp, item)

Arguments

resp A matrix of binary responses to a test, with persons as rows and items as columns.
item The item (column number in resp) to plot.

Value

None

Author(s)

Ivailo Partchev

Examples

erf(Scored, 3)

est

Estimate item parameters

Description

Estimate IRT item parameters using either ICL, BILOG, or ltm. Provides access to the most widely used options in these programs.

Usage

est(resp, model = "2PL", engine = "icl", omitsWrong = FALSE, nqp = 20,
est.distr = FALSE, nch = 5, a.prior = TRUE, b.prior = FALSE,
c.prior = TRUE, bilog.defaults = TRUE, rasch = FALSE,
run.name = "mymodel")
Arguments

resp A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data

model The IRT model: "1PL", "2PL", or "3PL". Default is "2PL".

engine One of "icl", "bilog", or "ltm". Default is "icl".

omitsWrong If TRUE, all omitted items will be recoded as wrong responses, otherwise they will be treated as missing (default)

nqp Number of quadrature points. Default is 20.

est.distr T if the probabilities of the latent distribution are to be estimated, F if a normal distribution is assumed. Default is F. Ignored when engine="ltm".

nch Number of choices in the original item formulation. Used to determine the prior for the asymptote when engine="bilog", model="3PL", and c.prior=T. Default is 5.

a.prior Whether a prior for the item discriminations is used. Ignored when model="1PL" or engine="ltm". Default is T.

b.prior Whether a prior for the item difficulties is used. Ignored when engine="ltm". Default is F.

c.prior Whether a prior for the asymptotes is used. Ignored when model="1PL" or model="2PL" or engine="ltm". Default is T.

bilog.defaults When engine="icl" and a prior is used, use the default priors in BILOG rather than the default priors in ICL. Ignored when engine="ltm". Default is T.

rasch When engine="bilog" and model="1PL" and "rasch"=T, the common value for discriminations is forced to 1, and the sum of the difficulties is 0. When engine="ltm" and model="1PL" and "rasch"=T, the common value for discriminations is forced to 1. Ignored in all other cases. Default is F.

run.name A (short) string used in the names of all files read or written by ICL or BILOG. Default is "mymodel". Change to something else to keep the outputs of ICL of BILOG for further use. Ignored when engine="ltm"

Details

Estimate the parameters of an IRT model defined in the most general case ("3PL") as

\[ P(U_{ij} = 1|\theta_i, a_j, b_j, c_j) = c_j + (1 - c_j) \frac{\exp(a_j(\theta_i - b_j))}{1 + \exp(a_j(\theta_i - b_j))} \]

where \( U_{ij} \) is a binary response given by person \( i \) to item \( j \), \( \theta_i \) is the value of the latent variable ("ability") for person \( i \), \( a_j \) is the discrimination parameter for item \( j \), \( b_j \) is the difficulty parameter for item \( j \), \( c_j \) is the asymptote for item \( j \).

Some authors prefer to represent the model with a logit \( 1.7a_j(\theta_i - b_j) \) rather than \( a_j(\theta_i - b_j) \). This option has been removed from \texttt{irtoys} as it is not supported by the remaining functions of the package.

In the 2PL model (model="2PL"), all asymptotes \( c_j \) are 0. In the 1PL model (model="1PL"), all asymptotes \( c_j \) are 0 and the discriminations \( a_j \) are equal for all items (and sometimes to 1).
Package `irtoys` provides a simple common interface to the estimation of item parameters with three different programs. It only accesses the most basic and widely used options in these programs. Each of the three programs has a much wider choice of options and capabilities, and serious users must still learn the corresponding syntax in order to access the advanced features. Even when models are fit "by hand", `irtoys` may be useful in plotting results, doing comparisons across programs etc.

Estimation of the more complex IRT models (2PL and 3PL) for some "difficult" data sets often has to use prior distributions for the item parameters. `irtoys` adopts the default behaviour of BILOG: no priors for \( b \) in any model, priors for \( a \) in the 2PL and 3PL models, priors for \( c \) in the 3PL model. This can be overridden by changing the values of \( \text{aprior} \), \( \text{bprior} \), and \( \text{cprior} \).

If priors are used at all, they will be the same for all items. Note that both ICL and BILOG can, at some additional effort, set different priors for any individual item. At default, the common priors are the BILOG defaults: normal \((\theta, 2)\) for \( b \), lognormal \((\theta, 0.5)\) for \( a \), and beta \((2\theta p + 1, 2(1-p) + 1)\) for \( c \); \( p \) is 1 over the number of choices in the original item formulations, which can be set with the parameter \( nch \), and is again assumed the same for all items.

When \( \text{engine} = \text{"icl"} \) and \( \text{bilog.defaults=F} \), any priors used will be the ICL default ones, and based on the 4-parameter beta distribution: beta \((1.01, 1.01, -6, 6)\) for \( b \), beta \((1.75, 3, 0, 3)\) for \( a \), and beta \((3.5, 4, 0, 0.5)\) for \( c \). When \( \text{engine} = \text{"ltm"} \), all commands involving priors are ignored.

`est` only works when some IRT software is installed. Package `ltm` is automatically loaded. ICL can be downloaded from [www.b-a-h.com](http://www.b-a-h.com). Alternative links are provided in the help screen for this package. BILOG is commercial software sold by SSI — see [www.ssicentral.com](http://www.ssicentral.com) for further detail. On Windows, make sure that the executable files (icl.exe for ICL, blm1.exe, blm2.exe, and blm3.exe, for BILOG) are located in directories that are included in the PATH variable.

### Value

A list with three elements: \( \text{est} \) and \( \text{se} \) are matrices containing the parameter estimates and their standard errors, correspondingly. Each matrix has three columns: [1] item discrimination \( a \), [2] item difficulty \( b \), and [3] asymptote \( c \). For the 1PL and 2PL models, all asymptotes are equal to 0; for the 1PL, the discriminations are all equal but not necessarily equal to 1. The third element, \( \text{vcm} \), is a list of variance-covariance matrices for the item parameters. Currently, \( \text{vcm} \) is only used in function `irfplot`. When ICL is used as estimation engine, \( \text{se} \) and \( \text{vcm} \) are NULL as ICL does not compute standard errors for the item parameter estimates.

### Author(s)

Ivailo Partchev

### References

Bradley A. Hanson (2002), ICL: IRT Command Language. [www.b-a-h.com](http://www.b-a-h.com)

Dimitris Rizopoulos (2006). Ltm: Latent Trait Models under IRT. [cran.r-project.org](http://cran.r-project.org)

Examples

```r
p.1pl <- est(Scored, model="1PL", engine="ltm")
p.2pl <- est(Scored, model="2PL", engine="ltm")
```

---

**iif**  
*Item information function*

**Description**

The item information function (IIF) for the 3PL model can be computed as

\[
I(\theta) = a^2 \frac{Q(\theta)}{P(\theta)} \left[ \frac{P(\theta) - c}{1 - e} \right]^2,
\]

where \( \theta \) is the value of the latent variable for a person, \( a \) is the discrimination parameter for the item, \( P \) is the IRF for the person and item, and \( Q = 1 - P \). For the 1PL and 2PL models, the expression reduces to \( a^2 PQ \).

**Usage**

```r
iif(ip, items = NULL, x = NULL)
```

**Arguments**

- **ip**: Item parameters: the output of `est`, or a 3-column matrix corresponding to its first element, `est`.
- **items**: The item(s) for which the information function is computed. If `NULL` (the default), `iir` for all items will be returned.
- **x**: The values of the latent variable (\( \theta \) in the equation above), at which the IIF will be evaluated. If not given, 99 values spaced evenly between -4 and +4 will be used, handy for plotting.

**Details**

A common use of this function would be to obtain a plot of the IIF.

**Value**

A list of:

- **x**: A copy of the argument `x`
- **f**: A matrix containing the IIF values: persons (values of `(x)`) as rows and items as columns
interactionModel

Author(s)
Ivailo Partchev

See Also
plot.iif, irf

Examples

plot(iif(Scored2pl, items=1:3))

-------------------------------------------------------------------------

interactionModel       Interaction model
-------------------------------------------------------------------------

Description
Estimates the parameters of the interaction model. The plots can be produced with the appropriate plotting method.

Usage
interactionModel(x, eps = 0.1, rasch = FALSE)

Arguments

x A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data
eps Convergence criterion, default is 1e-1
rasch if TRUE, the Rasch model will be estimated instead.

Value
A matrix

Author(s)
Ivailo Partchev, using C code and theory by Gunter Maris

References
Description

Returns the item response function of the 3PL (1PL, 2PL) model, the i.e. the probabilities defined by

\[ P(U_{ij} = 1|\theta_i, a_j, b_j, c_j) = c_j + (1 - c_j) \frac{\exp(a_j(\theta_i - b_j))}{1 + \exp(a_j(\theta_i - b_j))} \]

where \( U_{ij} \) is a binary response given by person \( i \) to item \( j \), \( \theta_i \) is the value of the latent variable ("ability") for person \( i \), \( a_j \) is the discrimination parameter for item \( j \), \( b_j \) is the difficulty parameter for item \( j \), \( c_j \) is the asymptote for item \( j \). Some authors call the IRF "the item characteristic curve".

Usage

\[ \text{irf}(\text{ip}, \text{items} = \text{NULL}, \text{x} = \text{NULL}) \]

Arguments

- \( \text{ip} \): Item parameters: the output of \text{est}, or a 3-column matrix corresponding to its first element, \text{est}.
- \( \text{items} \): The item(s) for which \text{irf} is computed. If \text{NULL} (the default), \text{irf} for all items will be returned.
- \( x \): The values of the latent variable (\( \theta \) in the equation above), at which the IRF will be evaluated. If not given, 99 values spaced evenly between -4 and +4 will be used, handy for plotting.

Details

In the 2PL model (\text{model}="2PL"), all asymptotes \( c_j \) are 0. In the 1PL model (\text{model}="1PL"), all asymptotes \( c_j \) are 0 and the discriminations \( a_j \) are equal for all items (and sometimes to 1).

A common use of this function would be to obtain a plot of the IRF.

Value

A list of:
- \( x \): A copy of the argument \( x \)
- \( f \): A matrix containing the IRF values: persons (values of \( x \)) as rows and items as columns

Author(s)

Ivailo Partchev
Description

Produces a plot that compares the estimated trace line (item response function) for an item with the data. The data is represented with a non-parametric trace line computed as in \texttt{tgf} (not \texttt{npp}). Approximate confidence intervals for the parametric line are also shown.

Usage

\begin{verbatim}
irfPlot(resp, ip, x = NULL, item, main = "Parametric vs non-parametric trace line", use.sm = FALSE)
\end{verbatim}

Arguments

- \texttt{resp} A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data
- \texttt{ip} Item parameters: the object returned by function \texttt{est} – note that the complete object is required since the standard errors play an important part.
- \texttt{x} A vector of abilities, as long as there are persons, against which to plot probabilities of a correct response. Normally these would be the output of \texttt{qrs}, and these are provided by default when \texttt{x} is NULL. However, the user may want to use something else, or pass the same \texttt{qrs} output to several plots (various calls to \texttt{qrs} provide slightly different output because of the random breaking of ties.)
- \texttt{item} The item for which a plot is requested (column of \texttt{resp}).
- \texttt{main} The main title of the plot.
- \texttt{use.sm} When \texttt{TRUE}, empirical data will be shown with a regression estimated by \texttt{sm}, otherwise the Bayes theorem will be used. Default is \texttt{FALSE}.

Details

Comparing the outputs of \texttt{npp} and \texttt{irf} has at least two drawbacks: (i) package \texttt{sm} is used as a black box, and (ii) the confidence intervals are drawn around the data (even though represented by the nonparametric regression) rather than the parametric curve. In this function, the parametric curve is shown in red, with a 68 The non-parametric curve representing the data is shown in black. It is computed by applying Bayes theorem to the density of ability. The standard error of the parametric curve is computed by applying the delta theorem on the standard error of the item parameters, as computed by the underlying software (currently only available with ICL).
Author(s)

Ivailo Partchev

See Also

irf, npp, tgf

Examples

```r
# a plot for item 5
irfPlot(Scored, ip=b3, item=4)
```

---

**itf**

*Test item fit*

Description

Returns a statistic of item fit together with its degrees of freedom and p-value. Optionally produces a plot.

Usage

```r
itf(resp, ip, item, stat = "1r", theta, standardize = TRUE, mu = 0,
    sigma = 1, bins = 9, breaks = NULL, equal = "count", type = "means",
    do.plot = TRUE, main = "Item fit")
```

Arguments

- `resp`: A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data
- `ip`: Item parameters: the object returned by `est`, or the equivalent of its first part.
- `item`: A single number pointing to the item (column of `resp`, row of `ip`), for which fit is to be tested
- `stat`: The statistic to be computed, either "chi" or "1r". Default is "1r". See details below.
- `theta`: A vector containing some viable estimate of the latent variable for the same persons whose responses are given in `resp`. If not given (and `group` is also missing), WLE estimates will be computed from `resp` and `ip`.
- `standardize`: Standardize the distribution of ability estimates?
- `mu`: Mean of the standardized distribution of ability estimates
- `sigma`: Standard deviation of the standardized distribution of ability estimates
- `bins`: Desired number of bins (default is 9)
- `breaks`: A vector of cutpoints. Overrides `bins` if present.
equal
Either "width" for bins of equal width, or "count" for bins with roughly counts
of observations. Default is "quant"

type
The points at which itf will evaluate the IRF. One of "mids" (the mid-point of
each bin), "meds" (the median of the values in the bin), or "means" (the mean
of the values in the bin). Default is "means".
do.plot
Whether to do a plot
main
The title of the plot if one is desired

Details
Given a long test, say 20 items or more, a large-test statistic of item fit could be constructed by
dividing examinees into groups of similar ability, and comparing the observed proportion of correct
answers in each group with the expected proportion under the proposed model. Different statistics
have been proposed for this purpose.
The chi-squared statistic
\[ X^2 = \sum_g \left( N_g \frac{(p_g - \pi_g)^2}{\pi_g (1 - \pi_g)} \right), \]
where \( N_g \) is the number of examinees in group \( g \), \( p_g = r_g / N_g \), \( r_g \) is the number of correct responses
to the item in group \( g \), and \( \pi_g \) is the IRF of the proposed model for the median ability in group \( g \),
is attributed by Embretson & Reise to R. D. Bock, although the article they cite does not actually
mention it. The statistic is the sum of the squares of quantities that are often called "Pearson
residuals" in the literature on categorical data analysis.
BILOG uses the likelihood-ratio statistic
\[ X^2 = 2 \sum_g \left[ r_g \log \frac{p_g}{\pi_g} + (N_g - r_g) \log \frac{(1 - p_g)}{(1 - \pi_g)} \right], \]
where \( \pi_g \) is now the IRF for the mean ability in group \( g \), and all other symbols are as above.

Both statistics are assumed to follow the chi-squared distribution with degrees of freedom equal to
the number of groups minus the number of parameters of the model (eg 2 in the case of the 2PL
model). The first statistic is obtained in itf with stat=\"chi\", and the second with stat=\"1lr\" (or
not specifying stat at all).
In the real world we can only work with estimates of ability, not with ability itself. irtoys allows
use of any suitable ability measure via the argument theta. If theta is not specified, itf will
compute EAP estimates of ability, group them in 9 groups having approximately the same number
of cases, and use the means of the ability estimates in each group. This is the approximate behaviour
of BILOG.

If the test has less than 20 items, itf will issue a warning. For tests of 10 items or less, BILOG has
a special statistic of fit, which can be found in the BILOG output. Also of interest is the fit in 2- and
3-way marginal tables in package ltm.

Value
A vector of three numbers:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistic</td>
<td>The value of the statistic of item fit</td>
</tr>
<tr>
<td>DF</td>
<td>The degrees of freedom</td>
</tr>
<tr>
<td>P-value</td>
<td>The p-value</td>
</tr>
</tbody>
</table>
mlebme

Maximum likelihood and Bayes Modal estimation of ability

Description

Estimates the value of the latent variable ("ability") for each person by direct optimization

Usage

mlebme(resp, ip, mu = 0, sigma = 1, method = "ML")

Arguments

- **resp**: A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data
- **ip**: Item parameters: the object returned by `est`.
- **mu**: Mean of the apriori distribution. Ignored when `method="ML"`. Default is 0.
- **sigma**: Standard deviation of the apriori distribution. Ignored when `method="ML"`. Default is 1.
- **method**: "ML" for maximum likelihood or "BM" for Bayes Modal estimation. Default is "ML", in which case any values for `mu` and `sigma` will be ignored.

Value

A matrix with the ability estimates in column 1 and their standard errors of measurement (SEM) in column 2, and the number of non-missing responses in column 3
normal.qu

Author(s)
Ivailo Partchev

See Also
eap

Examples

th.mle <- mleme(resp=Scored, ip=Scored2pl)

normal.qu

Normal quadrature points and weights

Description
Quadrature points and weights based on the Normal distribution. Quadrature objects are used when estimating abilities with eap and for some of the scaling methods in sca.

Usage

normal.qu(n = 15, lower = -4, upper = 4, mu = 0, sigma = 1,
scaling = "points")

Arguments

n
Number of quadrature points
lower
Lower boundary
upper
Upper boundary
mu
Mean
sigma
Standard deviation
scaling
Determines the way in which non-default values of mu and sigma are applied. When scaling="points", the quadrature points are rescaled, otherwise the quadrature weights are adapted.

Value
A list of:
quad.points
A vector of n quadrature points
quad.weights
A vector of the corresponding quadrature weights

Author(s)
Ivailo Partchev
See Also

read.quicl, eap, sca

Examples

quad <- normal.qu(n=20)

npp  Non-parametric characteristic curves

Description

A plotting routine producing non-parametric analogues of the IRF not unlike those in Jim Ramsay’s TestGraf program. The curves are produced by a kernel binomial regression of the actual responses to an item on some estimates of the latent variable, by courtesy of package sm.

Usage

npp(resp, x, items, from = -4, to = 4, co = 1,
    main = "Non-parametric response function", add = FALSE, bands = FALSE,
    label = FALSE)

Arguments

resp    A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data
x       The values of the latent variable ("ability") for the same persons whose responses are given in resp. If not given, function qrs will be plugged in, which is the approach of TestGraf
items   An index to the items (columns of resp) to be shown on the plot. If not given, all items will be plotted.
from    Lower limit for ability on the plot. Default is -4.
to      Upper limit for ability on the plot. Default is 4.
co      The colour of the curves. Default is 1 for black. Use co=NA to plot each curve in a different colour.
main    The main title of the plot, given that add=F.
add     When add=T, the curve is added to a plot, otherwise a new plot is started. Default is F.
bands   When bands=T, confidence bands are added.
label   When label=T, individual curves will be labeled with the item number.
Author(s)
Ivailo Partchev

References
James O. Ramsay (2000). TestGraf: A program for the graphical analysis of multiple choice test and questionnaire data. McGill University, Montreal, Canada

See Also
qrs, irf, plot.irf

Examples

# plot items 1:5 in different colours, label
npp(Scored, items=1:5, co=NA, label=TRUE)

# For item 7, compare npp with the 2PL parametric IRF
npp(Scored, items=7, bands=TRUE)
plot(irf(ip=Scored2pl, items=7), co=3, add=TRUE)

plot.iif

A plot method for item information functions

Description
Useful for plotting item information functions. The x argument of iif should better be left out unless something special is required.

Usage

## S3 method for class 'iif'
plot(x, add = FALSE, main = "Item information function",
     co = 1, label = FALSE, ...)

Arguments

x An object produced by function iif
add When add=T, the IIF is added to a plot, otherwise a new plot is started. Default is F.
main The main title of the plot, given that add=F.
co The colour of the IIF curve. Default is 1 for black. Use co=NA to plot each IIF in a different colour.
label When label=T, individual curves will be labeled with the item number.
... Any additional plotting parameters
### plot.imp

A plot method for the interaction model

#### Description

Plot the item-total regressions fit by the interaction (or Rasch) model

#### Usage

```
## S3 method for class 'imp'
plot(x, items = NULL, shade = 10, highlight = -Inf,
     add = FALSE, main = "Item-total regression", label = FALSE, ...)
```

#### Arguments

- **x**: An object produced by function `interactionModel`
- **items**: The items to plot (column numbers). If NULL, all items will be plotted
- **shade**: The part of the probability mass for the sum scores to shade out, shown as percentage. Default is 10: shade the most extreme 10 Ignored when `add=TRUE`.
- **highlight**: Cutpoint for the interaction parameter below which a regression line will be highlighted in red. Default is -Inf (do not highlight).
- **add**: When `add=TRUE`, the graph is added to a plot, otherwise a new plot is started. Default is FALSE.
- **main**: The main title of the plot, given that `add=FALSE`.
- **label**: When `label=TRUE`, individual curves will be labeled with the item number.
- **...**: Any additional plotting parameters

#### Author(s)

Ivailo Partchev, using theory and C code by Gunter Maris
plot.irf

See Also

interactionModel

Examples

plot(interactionModel(Scored), highlight=-.3)

Description

Useful for plotting item response functions. The x argument of irf should better be left out unless something special is required.

Usage

## S3 method for class 'irf'
plot(x, add = FALSE, main = "Item response function",
    co = 1, label = FALSE, ...)

Arguments

x An object produced by function irf

add When add=T, the IRF is added to a plot, otherwise a new plot is started. Default is F.

main The main title of the plot, given that add=F.

col The colour of the IRF curve. Default is 1 for black. Use col=NA to plot each IRF in a different colour.

label When label=T, individual curves will be labeled with the item number.

... Any additional plotting parameters

Author(s)

Ivailo Partchev

See Also

irf
Examples

```r
# plot IRF for all items in red, label with item number
plot( irf(Scored2pl), co="red", label=TRUE)

# plot IRF for items 2, 3, and 7 in different colours
plot( irf( Scored2pl, items=c(2,3,7)), co=NA)
```

---

**plot.tif**  
*An plot method for test information functions*

**Description**

Useful for plotting test information functions. The `x` argument of `tif` should better be left out unless something special is required.

**Usage**

```r
## S3 method for class 'tif'
plot(x, add = FALSE, main = "Test information function",
     co = 1, ...)
```

**Arguments**

- `x`  
  An object produced by function `tif`

- `add`  
  When `add=TRUE`, the TIF is added to a plot, otherwise a new plot is started. Default is `FALSE`.

- `main`  
  The main title of the plot, given that `add=FALSE`.

- `co`  
  The colour of the TIF curve. Default is 1 for black. Use `co=NA` to plot each TIF in a different colour.

- `...`  
  Any additional plotting parameters

**Author(s)**

Ivailo Partchev

**See Also**

`tif`

**Examples**

```r
plot( tif( Scored2pl))
```
Description

Useful for plotting test response functions. The x argument of trf should better be left out unless something special is required.

Usage

```r
## S3 method for class 'trf'
plot(x, add = FALSE, main = "Test response function",
     co = 1, ...)  
```

Arguments

- `x`  
  An object produced by function `trf`

- `add`  
  When add=TRUE, the IRF is added to a plot, otherwise a new plot is started. Default is FALSE.

- `main`  
  The main title of the plot, given that add=F.

- `co`  
  The colour of the TRF curve. Default is 1 for black. Use co=NA to plot each TRF in a different colour.

- `...`  
  Any additional plotting parameters

Author(s)

Ivailo Partchev

See Also

`trf`

Examples

```r
plot(trf(Scored2pl))
```
Quantiles of the ranked sum scores

Description
A rough estimate of the values of the latent variable ("ability"). The sum scores (number of correct responses) are ranked, breaking ties at random. The ranks are divided by the sample size + 1, and the corresponding quantiles of the standard Normal distribution are returned. Used as default in the non-parametric IRF plots produced by \texttt{npp} in analogy to Jim Ramsay’s TestGraf. Another possible use is in \texttt{itf}.

Usage
\begin{verbatim}
qrs(resp)
\end{verbatim}

Arguments
\begin{verbatim}
resp A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data
\end{verbatim}

Value
A one-column matrix of values

Author(s)
Ivailo Partchev

See Also
\begin{verbatim}
npp.itf
\end{verbatim}

Examples
\begin{verbatim}
sc <- qrs(Scored)
\end{verbatim}
**read.ip.bilog**

**Read in parameter estimates**

**Description**

From BILOG output, read in estimates of item parameters. Invoked automatically when the model is estimated via irtoys. If that is not the case, file must be a file produced with the `SAVE PARm` file; command in BILOG.

**Usage**

```r
code; read.ip.bilog(file)
```

**Arguments**

- `file` File name

**Value**

A list with three elements, `est`, `se`, `vcm` for the parameter estimates, their standard errors, and the variance-covariance matrix per item. The first two are matrices with one row per item, and three columns: `[1]` item discrimination $a$, `[2]` item difficulty $b$, and `[3]` asymptote $c$. For the 1PL and 2PL models, all asymptotes are equal to 0; for the 1PL, the discriminations are all equal but not necessarily equal to 1.

**Author(s)**

Ivailo Partchev

---

**read.ip.icl**

**Read in parameter estimates**

**Description**

From ICL output, read in estimates of item parameters. Invoked automatically when the model is estimated via irtoys. If that is not the case, file must be a file produced with the `write_item_param` file command in ICL.

**Usage**

```r
code; read.ip.icl(file)
```

**Arguments**

- `file` File name
Value

A list with three elements, est, se, vcm for the parameter estimates, their standard errors, and the variance-covariance matrix per item. Because ICL does not compute standard errors, se and vcm will be NULL. est is a matrix with one row per item, and three columns: [1] item discrimination \( a \), [2] item difficulty \( b \), and [3] asymptote \( c \). For the 1PL and 2PL models, all asymptotes are equal to 0; for the 1PL, the discriminations are all equal but not necessarily equal to 1.

Author(s)

Ivailo Partchev

Description

From ICL output, read in estimates of item parameters. file must be a file produced with the \texttt{write_latent_dist} file command in ICL. Quadrature objects are used when estimating abilities with \texttt{eap} and for some of the scaling methods in \texttt{sca}.

Usage

\texttt{read.qu.icl(file)}

Arguments

\begin{itemize}
  \item \texttt{file} File name
\end{itemize}

Value

A list of:

\begin{itemize}
  \item \texttt{quad.points} A vector of quadrature points
  \item \texttt{quad.weights} A vector of the corresponding quadrature weights
\end{itemize}

Author(s)

Ivailo Partchev

See Also

\texttt{normal.qu, eap, sca}
read.resp  

Read responses from a file

Description

Reads responses to a questionnaire from a text file

Usage

read.resp(file, na = ".")

Arguments

- **file**: File name
- **na**: The symbol used to represent missing data

Details

Included for those who are too faint-hearted to write as.matrix(read.table(file, head=F)). Of course, data can be entered into R in many other ways.

The data values in the file must be separated with blanks.

Responses are the empirical data used in IRT. Note that irtoys deals with models for dichotomous data, and typically expects data consisting of zeroes and ones, without any missing values (non-responses are considered as wrong responses). In fact, there are only two commands in irtoys that accept other kinds of data: sco and tgf.

read.resp does accept missing data and values other than 0 and 1. Use sco and a key to score multiple choice responses to 0/1. If you have dichotomous data that contains NAs, you can use sco without a key to change all NA to 0.

Value

A matrix, typically of zeroes and ones, representing the correct or wrong responses given by persons (rows) to items (columns).

Author(s)

Ivailo Partchev

See Also

sco, tgf,
Examples

```r
## Not run:
r <- read.resp("c:/myfiles/irt.dat")
## End(Not run)
```

```
rim

Item-total regressions for the Rasch vs. the interaction model
```

Description

Compare the item-total regressions fit by the Rasch model and the interaction model, for one, several, or all items in the test.

Usage

```r
rim(x, items = NULL, showData = FALSE, shade = 10, ncol = 3, ...)
```

Arguments

- `x`: A matrix of scored responses (persons in rows, items in columns)
- `items`: The items to plot (column numbers). If NULL, all items will be plotted
- `showData`: If TRUE, the observed proportion correct at each sum score will be shown on the plot. Default is FALSE (show only the regressions)
- `shade`: The part of the probability mass for the sum scores to shade out, shown as percentage. Default is 10: shade the most extreme 10. Ignored when `add=TRUE`
- `ncol`: When plotting multiple items, the user can suggest the number of columns in the matrix of plots (up to 3, will be adjusted if necessary)
- `...`: Any additional plotting parameters

Author(s)

Ivailo Partchev, using theory and C code by Gunter Maris

See Also

- interactionModel

Examples

```r
rim(x, items=2)
```
Description

Linearly transform a set of IRT parameters to bring them to the scale of another set of parameters. Four methods are implemented: Mean/Mean, Mean/Sigma, Lord-Stocking, and Haebara.

Usage

sca(old.ip, new.ip, old.items, new.items, old.qu = NULL, new.qu = NULL, method = "MS", bec = FALSE)

Arguments

old.ip A set of parameters that are already on the desired scale
new.ip A set of parameters that must be placed on the same scale as old.ip
old.items A vector of indexes pointing to those items in old.ip that are common to both sets of parameters
new.items The indexes of the same items in new.ip
old.qu A quadrature object for old.ip, typically produced by the same program that estimated old.ip. Only needed if method="LS" or method="HB"
new.qu A quadrature object for new.ip, typically produced by the same program that estimated new.ip. Only needed if method="HB"
method One of "MM" (Mean/Mean), "MS" (Mean/Sigma), "SL" (Stocking-Lord), or "HB" (Haebara). Default is "MS"
bec Use back-equating correction? When TRUE, the Stocking-Lord or Hebaera procedures will be adjusted for back-equating (see Hebaera, 1980). Ignored when method is MM or MS. Default is FALSE.

Value

A list of:
slope The slope of the linear transformation
intercept The intercept of the linear transformation
scaled.ip The parameters in new.ip transformed to a scale that is compatible with old.ip

Author(s)

Ivailo Partchev and Tamaki Hattori
## sco

**Score a multiple choice test**

### Description

Given a key, score a multiple choice test, i.e. recode the original choices to right (1) or wrong (0). Missing responses are treated as wrong.

### Usage

```r
sco(choices, key, na.false = FALSE)
```

### Arguments

- `choices`: The original responses to the items in the test: persons as rows, items as columns. May contain NA.
key

A vector containing the key (correct answers) to the items in choices. If not given, the function will check if all data are either 0, 1, or NA: if yes, NA are recoded as 0, else an error message is returned.

na.false

Recode non-responses to false responses?

Value

A matrix of responses scored 0=wrong 1=correct, and possibly NA

Author(s)

Ivailo Partchev

Examples

res <- sco(Unscored, key=c(2,3,1,1,4,1,2,1,2,3,3,4,3,4,2,2,4,3))

Scored

Binary (true/false) responses to a test

Description

Real-life data set containing the responses to a test, scored as true or false.

Format

A data set with 472 persons and 18 items.

Scored2pl

Example item parameters

Description

Item parameter estimates for the 2PL model, estimated with \texttt{ltm} from the example data set \texttt{Scored}. These are provided as a check, and to speed up the examples for the various functions in the package.

Format

The object returned by \texttt{est}
scoreMetric | *Sum score metric*

**Description**

From a set of estimated item parameters, compute an item response function in sum score metric, i.e., the probabilities of a correct response predicted by the model at abilities corresponding to the possible (raw) sum scores.

**Usage**

```r
scoreMetric(ip)
```

**Arguments**

- `ip`: Item parameters: the output of `est`, or a 3-column matrix corresponding to its first element, `est`.

**Value**

An object with the same structure as the output of `irf`.

**Author(s)**

Ivailo Partchev

---

scp | *Plot observed and predicted scores against ability*

**Description**

Produces a plot of IRT true scores (test response function at the estimated ability) with a confidence band (plus/minus standard error). The observed sum scores are shown in red.

**Usage**

```r
scp(resp, ip, theta = NULL)
```

**Arguments**

- `resp`: A matrix of binary responses to a test, with persons as rows and items as columns.
- `ip`: Item parameters: the output of `est`, or a 3-column matrix corresponding to its first element, `est`.
- `theta`: An object containing ability estimates, as output by function `mlebme` or `eap`. If `NULL`, MLE will be estimated from `resp` and `ip`.

---
**Value**

None

**Author(s)**

Ivailo Partchev

**See Also**

mlebme, eap, tsc, trf

**Examples**

```r
scp(Scored, Scored2pl)
```

---

### Description

Simulate responses from the 1PL, 2PL, or 3PL model

### Usage

```r
sim(ip, x = NULL)
```

### Arguments

- **ip**
  - Item parameters: a matrix with one row per item, and three columns: [1] item discrimination \(a\), [2] item difficulty \(b\), and [3] asymptote \(c\). Can also use the object returned by `est`.

- **x**
  - A vector of values of the latent variable ("abilities").

### Value

A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data

### Author(s)

Ivailo Partchev

### Examples

```r
pa <- cbind(runif(20,.8,2), runif(20,-2.4,2.4), rep(0,50))
rs <- sim(ip=pa, x=rnorm(1000))
```
tet Approximate tetchoric correlation matrix

Description
Matrix of tetchoric correlations using the approximation by Bonett and Price (2005).

Usage
tet(d)

Arguments
d a matrix (or data frame, which will be converted to a matrix) containing only zeroes an ones. NAs are not allowed.

Value
A matrix of approximate tetchoric correlations.

Author(s)
Ivailo Partchev

References

Examples
tets <- tet(Scored)

tgf Non-parametric option curves

Description
A plotting function producing non-parametric analogues of the IRF for each option in a multiple choice item not unlike those in Jim Ramsay’s TestGraf program.

Usage
tgf(choices, key, item, main = "Non-parametric response function", co = 1, label = FALSE)
**Arguments**

- **choices**: A matrix of responses to multiple-choice items: persons as rows, items as columns. As a rare exception in irtoys, responses must not be recoded to 0/1, and there may be missing responses.
- **key**: A vector containing the key (correct answers) to the items in choices.
- **item**: A single number pointing to the item (column of choices) to plot.
- **main**: The main title of the plot, given that add=F.
- **co**: The colour of the curves. Default is 1 for black. Use co=NA to plot each curve in a different colour.
- **label**: When label=T, individual curves will be labeled with the item number.

**Author(s)**

Ivailo Partchev

**References**

James O. Ramsay (2000). TestGraf: A program for the graphical analysis of multiple choice test and questionnaire data. McGill University, Montreal, Canada

**See Also**

- `qrs`, `irf`, `plot.irf`

**Examples**

```r
key=c(2,3,1,1,4,1,2,1,2,3,3,4,3,4,2,2,4,3)
tgf(choices=Unscored, key=key, item=4, co=NA, label=TRUE)
```

---

**tia**  
*Elementary test-item analysis*

**Description**

Elementary analysis of the items in a test and the test sumscores based on Classical Test Theory.

**Usage**

`tia(choices, key, ...)`
Arguments

choices The original responses to the items in the test: persons as rows, items as columns. May contain NA.

key A vector containing the key (correct answers) to the items in choices. If not given, the function will check if all data are either 0, 1, or NA: if yes, NA are recoded as 0, else an error message is returned.

... Other parameters that may be passed to sco or cov

Value

A list with three elements:

- **testlevel** A list of statistics at test level (currently, only Cronbach’s alpha, may be extended in future)
- **itemlevel** A matrix showing, for each item, the proportion of correct responses, the correlation with the sum score, and the alpha that the test would have if the item were dropped.
- **optionlevel** A matrix showing, for each possible choice in the multiple-choice item, the proportion of responses given, and the correlation with the sum score for the test (including the item). The correct response is highlighted with asterisks.

Author(s)

Ivailo Partchev

Examples

```r
itemsum <- tia(Unscored, key=c(2,3,1,1,4,1,2,1,2,3,4,3,4,2,4,3))
```

Description

Returns the test information function (TIF) of the 3PL (1PL, 2PL) model. The TIF is the sum of the item information functions (IIF) in a test, and indicates the precision of measurement that can be achieved with the test at any value of the latent variable, being inversely related to measurement variance.

Usage

```r
tif(ip, x = NULL)
```
Arguments

ip  Item parameters: the output of `est`.

x  The values of the latent variable (θ in the equation above), at which the TIF will be evaluated. If not given, 99 values spaced evenly between -4 and +4 will be used, handy for plotting.

Details

A common use of this function would be to obtain a plot of the TIF.

Value

A list of:

x  A copy of the argument x

f  A vector containing the TIF values

Author(s)

Ivailo Partchev

See Also

`plot.tif`, `iif`

Examples

```r
plot(trf(Scored2pl))
```

---

**Description**

Returns the test response function (TRF) of the 3PL (1PL, 2PL) model. The TRF is the sum of the item response functions (IRF) in a test, and represents the expected score as a function of the latent variable θ.

**Usage**

```r
trf(ip, x = NULL)
```
Arguments

ip  
Item parameters: the output of est, or a 3-column matrix corresponding to its first element, est.

x  
The values of the latent variable (θ in the equation above), at which the IRF will be evaluated. If not given, 99 values spaced evenly between -4 and +4 will be used, handy for plotting.

Details

A common use of this function would be to obtain a plot of the TRF.

Value

A list of:

x  
A copy of the argument x

f  
A vector containing the TRF values

Author(s)

Ivailo Partchev

See Also

plot.trf, irf

Examples

plot(trf(Scored2pl))

desc

True scores with standard errors

Description

Computes the IRT true scores (test response function at the estimated ability) and an estimate of their standard error via the delta theorem, treating item parameters as known.

Usage

tsc(ip, theta)

Arguments

ip  
Item parameters: the output of est, or a 3-column matrix corresponding to its first element, est.

theta  
An object containing ability estimates, as output by function mlebme or eap
Value

A matrix with the true scores in column 1, and their standard errors of measurement (SEM) in column 2

Author(s)

Ivailo Partchev

See Also

mlebme, eap, trf

Examples

```r
th <- mlebme(resp=Scored, ip=Scored2pl)
tsc(Scored2pl, th)
```

Unscored  | Original, unscored multiple-choice responses to a test

Description

Real-life data set containing the responses to a test, before they have been recoded as true or false. Can be used with only two functions in the package: scoreResponses and tgf. All other functions expect binary data, which can be produced with scoreResponses.

Format

A data set with 472 persons and 18 items. Each item has 4 possible answers, of which only one is true. There are many NA, which can be treated as wrong responses.

wle  | Bias-corrected (Warm’s) estimates of ability

Description

Weighted likelihood estimates (WLE) of ability, designed to remove the first order bias term from the ML estimates. WLE are finite for response patterns consisting of either uniformly wrong or uniformly correct responses.

Usage

```r
wle(resp, ip)
```
Arguments

- **resp**
  
  A matrix of responses: persons as rows, items as columns, entries are either 0 or 1, no missing data

- **ip**
  
  Item parameters: the object returned by `est`.

Value

A matrix with the ability estimates in column 1, and their standard errors of measurement (SEM) in column 2, and the number of non-missing responses in column 3

Author(s)

Ivailo Partchev

References


See Also

- mlebme, eap

Examples

```
th.bce <- wle(resp=Scored, ip=Scored2pl)
```
Index

*Topic IO
read.resp, 29

*Topic datasets
b1, 6
b2, 6
b3, 6
Scored, 33
Scored2pl, 33
Unscored, 41

*Topic data
read.ip.bilog, 27

*Topic models
ability, 4
api, 5
dpv, 7
eap, 8
erf, 9
est, 9
iif, 12
interactionModel, 13
irf, 14
irtoys-package, 3
itf, 16
mlebme, 18
normal.qu, 19
npp, 20
plot.iif, 21
plot.imp, 22
plot.irf, 23
plot.tif, 24
plot.trf, 25
qrs, 26
read.ip.icl, 27
read.qu.icl, 28
rim, 30
sca, 31
sco, 32
scoreMetric, 34
scp, 34
sim, 35
tet, 36
tgf, 36
tia, 37
tif, 38
trf, 39
tsc, 40
wle, 41
ability, 4
api, 5
b1, 6
b2, 6
b3, 6
dpv, 5, 7
eap, 5, 7, 8, 18–20, 28, 35, 41, 42
erf, 9
est, 9
iif, 12, 22, 39
interactionModel, 13, 23, 30
irf, 13, 14, 16, 21, 23, 37, 40
irfPlot, 15
irtoys (irtoys-package), 3
irtoys-package, 3
itf, 16, 26
mlebme, 5, 7, 8, 18, 35, 41, 42
normal.qu, 8, 19, 28
npp, 16, 20, 26
plot.iif, 13, 21
plot.imp, 22
plot.irf, 15, 21, 23, 37
plot.tif, 24, 39
plot.trf, 25, 40
qrs, 5, 18, 21, 26, 37
read.ip.bilog, 27
read.ip.icl, 27
read.qu.icl, 8, 20, 28
read.resp, 29
rim, 30
sca, 20, 28, 31
sco, 29, 32
Scored, 33
Scored2pl, 33
scoreMetric, 34
scp, 34
sim, 35
tet, 36
tgf, 16, 29, 36
tia, 37
tif, 24, 38
trf, 25, 35, 39, 41
tsc, 35, 40
Unscored, 41
wle, 5, 41