Package ‘irtoys’

February 20, 2015

Type Package
Title Simple interface to the estimation and plotting of IRT models
Version 0.1.7
Date 2014-01-20
Author Ivailo Partchev
Maintainer Ivailo Partchev <partchev@gmail.com>
Description Provides 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, and a variety of functions useful with IRT models.
LazyLoad yes
LazyData yes
License GPL (>= 2)
Depends sm, ltm
Suggests MASS
Collate 'ability.R' 'estimate.R' 'itemfit.R' 'nonpar.R' 'readin.R'
   'scale.R' 'tracelines.R' 'utils.R' 'irtoys-package.R'
NeedsCompilation no
Repository CRAN
Date/Publication 2014-01-23 00:16:27

R topics documented:

irtoys-package .................................................. 2
api ................................................................... 4
dpv ................................................................. 5
eap ................................................................. 6
est .................................................................. 7
iif .................................................................. 9
irf .................................................................. 10
itf .................................................................. 11
mlebme ............................................................. 14
irtoys-package

Description

Package: irtoys
Type: Package
Version: 0.1.4
Date: 2011-06-22
License: GPL (>= 2)
LazyLoad: yes
LazyData: yes

Index

irtoys-package Estimate and plot IRT models for binary responses
Details

Provides a common interface to the estimation of item parameters in IRT models for binary responses with three different programs (ICL, BILOG, and ltm, and a variety of functions useful with IRT models.

The irtoys package contains a bunch of functions potentially useful to those teaching or learning Item Response Theory (IRT). Although there is no shortage of good IRT programs, those tend to have wildly different and often unwieldy user interfaces. Besides, no single program does everything one needs. Item parameters can be estimated with a program like ICL or BILOG, non-parametric approaches are implemented in TestGraf, transformation to a common scale needs ST, and so on. Some programs, such as ICL, have no graphical capabilities at all, while others offer stunning interactive graphics but refuse to output a Postscript file.

Package irtoys provides a common interface to some of the most basic functions in ICL, BILOG, and 's own ltm, some of the functionality of TestGraf and ST, and a variety of other functions. Those who want to take advantage of the full functionality of ICL, BILOG & Co. must still master their syntax.

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. 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 0.1.6, function est returns a list of two matrices: est contains the parameter estimates and is thus identical to the output in earlier versions, and se contains the standard errors, in a similar format. 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 \( 1.7a_j^2(\theta_i - b_j) \), function est now estimates item parameters only in the logistic metric.

Author(s)

Ivailo Partchev <partchev@gmail.com>

References

The Z3 appropriateness index

Description

Computes the Z3 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: a matrix with one row per item, and three columns: [.1] item discrimination \( a \), [.2] item difficulty \( b \), and [.3] asymptote \( c \).
- **theta**: A measure of ability, typically produced with mle, 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$est)
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: a matrix with one row per item, and three columns: [.1] item discrimination \( a \), [.2] item difficulty \( b \), and [.3] asymptote \( c \).
- `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.
- `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$est)
```
eap  

EAP estimation of ability

Description

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

Usage

eap(resp, ip, qu)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resp</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>ip</td>
<td>Item parameters: a matrix with one row per item, and three columns: [1] item discrimination ( a ), [2] item difficulty ( b ), and [3] asymptote ( c ).</td>
</tr>
<tr>
<td>qu</td>
<td>A quadrature object produced with normal.qu or read in with read.qu.icl</td>
</tr>
</tbody>
</table>

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.eap <- eap(resp=Scored, ip=Scored2pl$est, qu=normal.qu())
```
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

```r
est(resp, model = "2PL", engine = "icl", 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".
- **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="1tm". Default is T.
- **b.prior**: Whether a prior for the item difficulties is used. Ignored when engine="1tm". Default is F.
- **c.prior**: Whether a prior for the asymptotes is used. Ignored when model="1PL" or model="2PL" or engine="1tm". 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

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

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{irttools} 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 \texttt{irttools} 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", \texttt{irttools} 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. \texttt{irttools} 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 \texttt{a.prior}, \texttt{b.prior}, and \texttt{c.prior}.

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: \texttt{normal(0, 2)} for \( b \), \texttt{lognormal (0 , 0.5)} for \( a \), and \texttt{beta(20*p+1, 20*(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 \texttt{nch}, and is again assumed the same for all items.

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

\texttt{est} only works when some IRT software is installed. Package \texttt{ltm} is automatically loaded. ICL can be downloaded from \url{www.b-a-h.com}. BILOG is commercial software sold by SSI — see \url{www.ssicentral.com} for further detail. On Windows, make sure that the executable files (icl.exe for ICL, b1m1.exe, b1m2.exe, and b1m3.exe, for BILOG) are located in directories that are included in the \texttt{PATH} variable.

Value

A list with two elements, \texttt{est} and \texttt{se}, for the parameter estimates and their standard errors, correspondingly. Each element is a matrix with one row per item, and three columns: \texttt{[,1]} item discrimination \( a \), \texttt{[,2]} item difficulty \( b \), and \texttt{[,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. When ICL is used as estimation engine, \texttt{se} is NULL as ICL does not compute standard errors for the item parameter estimates.
Description

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

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

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

iif(ip, x = NULL)

Arguments

- **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:

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

Author(s)

Ivailo Partchev

See Also

- `plot.iif`
- `irf`

Examples

```r
plot(iif(Scored2pl$est[,1:3]))
```

---

**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**

```r
irf(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 \(c\).
- `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 (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).

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

See Also

plot.rif

Examples

```r
plot(irm(Scored2pl$est[1,]))
```

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 = "lr", 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
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), EAP 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 \( \text{itf} \) with \texttt{stat=“chi”}, and the second with \texttt{stat=“lr”} (or not specifying \texttt{stat} at all).

In the real world we can only work with estimates of ability, not with ability itself. \( \text{irtoys} \) allows use of any suitable ability measure via the argument \texttt{theta}. If \texttt{theta} is not specified, \( \text{itf} \) 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, \( \text{itf} \) 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 \( \text{ltm} \).

**Value**

A vector of three numbers:

- **Statistic**: The value of the statistic of item fit
- **DF**: The degrees of freedom
- **P-value**: The p-value

**Author(s)**

Ivailo Partchev

**References**


**See Also**

\( \text{eap}, \text{qrs} \)

**Examples**

\[ \text{fit} \leftarrow \text{itf}(\text{resp=Scored}, \text{ip=Scored2pl$est}, \text{item}=7) \]
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: a matrix with one row per item, and three columns: [1] item discrimination \(a\), [2] item difficulty \(b\), and [3] asymptote \(c\).
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

Author(s)

Ivailo Partchev

See Also

eap

Examples

th.mle <- mlebme(resp=Scored, ip=Scored2pl$est)
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.qu.icl, 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 \texttt{sm}.

Usage

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

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 \texttt{resp}. If not given, function \texttt{qrs} will be plugged in, which is the approach of TestGraf.
- **items**: An index to the items (columns of \texttt{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.
- **add**: When \texttt{add=T}, the curve is added to a plot, otherwise a new plot is started. Default is \texttt{F}.
- **main**: The main title of the plot, given that \texttt{add=F}.
- **co**: The colour of the curves. Default is 1 for black. Use \texttt{co=NA} to plot each curve in a different colour.
- **bands**: When \texttt{bands=T}, confidence bands are added.
- **label**: When \texttt{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
plot.iif

See Also

qrs, rfit, plot.rfit

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(rfit(ip=Scored2pl$est[7,]), co=3, add=TRUE)

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

Author(s)

Ivailo Partchev

See Also

iif
Examples

# plot IIF for all items in red, label with item number
plot(irf(Scored2pl$est), co="red", label=TRUE)
# plot IIF for items 2, 3, and 7 in different colours
plot(irf(Scored2pl$est[c(2,3,7),]), co=NA)

plot.irf

A plot method for item response functions

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.
co  The colour of the IRF curve. Default is 1 for black. Use co=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

# plot IIF for all items in red, label with item number
plot(irf(Scored2pl$est), co="red", label=TRUE)
# plot IIF for items 2, 3, and 7 in different colours
plot(irf(Scored2pl$est[c(2,3,7),]), co=NA)
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$est))
```
A plot method for test response functions

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=FALSE.
- `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$est))
```
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 npp in analogy to Jim Ramsay’s TestGraf. Another possible use is in itf.

Usage
qrs(resp)

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

Value
A one-column matrix of values

Author(s)
Ivailo Partchev

See Also
npp, itf

Examples
sc <- qrs(Scored)

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

read.ip.bilog(file)

Arguments

file File name

Value

A list with two elements, est and se, for the parameter estimates and their standard errors, correspondingly. Each element 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

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

read.ip.icl(file)

Arguments

file File name

Value

A list with two elements, est and se, for the parameter estimates and their standard errors, correspondingly. Because ICL does not compute standard errors, se 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
**read.qu.icl**

**Read in quadrature**

**Description**

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

**Usage**

```
read.qu.icl(file)
```

**Arguments**

- **file**
  - File name

**Value**

A list of:

- **quad.points**
  - A vector of quadrature points
- **quad.weights**
  - A vector of the corresponding quadrature weights

**Author(s)**

Ivailo Partchev

**See Also**

`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 tgp.

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, tgp,

Examples

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

sca

Linear transformation of the IRT scale

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

```r
csa(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 Haebara procedures will be adjusted for back-equating (see Haebara, 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

References


Examples

```
## Not run:
# a small simulation to demonstrate transformation to a common scale
# fake 50 2PL items
pa <- cbind(runif(50,.8,2), runif(50,-2.4,2.4), rep(0,50))
# simulate responses with two samples of different ability levels
r.1 <- sim(ip=pa[1:30,], x=rnorm(1000,-.5))
r.2 <- sim(ip=pa[21:50,], x=rnorm(1000,.5))
# estimate item parameters
p.1 <- est(r.1, engine="ltm")
p.2 <- est(r.2, engine="ltm")
# plot difficulties to show difference in scale
```
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
`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
```r
res <- sco(Unscored, key=c(2,3,1,1,4,1,2,1,2,3,3,4,3,4,2,2,4,3))
```
Scored

**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 ltm from the example data set Scored. These are provided as a check, and to speed up the examples for the various functions in the package.

**Format**

A list of two matrices: est contains the parameter estimates, and se contains the standard errors (see also est).

---

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**

```
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: a matrix with one row per item, and three columns: [1] item discrimination \(a\), [2] item difficulty \(b\), and [3] asymptote \(c\).
- **theta** An object containing ability estimates, as output by function mlebme or eap. If NULL, MLE will be estimated from resp and ip.
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 $c$.
- `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

**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 tetrachoric correlation matrix

Description
Matrix of tetrachoric 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 and ones. NAs are not allowed.

Value
A matrix of approximate tetrachoric correlations.

Author(s)
Ivailo Partchev

References

Examples
tetras <- tet(Scored)

tgp
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
tgp(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=T.

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

key=c(2,3,1,1,4,1,2,1,2,3,3,4,4,2,4,3)
tgp(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
testsum <- tia(Unscored, key=c(2,3,1,1,4,1,2,1,2,3,3,4,3,4,2,2,4,3))
```

---

**tif**  
*Test information function*

**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: a matrix with one row per item, and three columns: [,1] item discrimination $a$, [,2] item difficulty $b$, and [,3] asymptote $c$.
- **x** The values of the latent variable ($\theta$ 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.
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**

- `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.tif`, `iif`
**tsc**

**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**

```r
tsc(ip, theta)
```

**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 \(c\).

- `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$est)
tsc(Scored2pl$est, 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: `sco` and `npp`. All other functions expect binary data, which can be produced with `sco`.

**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**

`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: a matrix with one row per item, and three columns: 
  - [.1] item discrimination \( a \)
  - [.2] item difficulty \( b \)
  - [.3] asymptote \( c \).

**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 reponses in column 3

**Author(s)**

Ivailo Partchev

**References**

wle

See Also

mlebme, eap

Examples

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

*Topic IO
  read.resp, 23
*Topic datasets
  Scored, 27
  Scored2pl, 27
  Unscored, 34
*Topic data
  read.ip.bilog, 21
*Topic models
  api, 4
  dpv, 5
  eap, 5, 6, 13–15, 23, 28, 33, 35
  est, 7
  iif, 9, 17, 32
  irf, 10, 17, 18, 30, 33
  irtoys (irtoys-package), 2
  irtoys-package, 2
  itf, 11, 21
  mlebme, 5, 6, 14, 28, 33, 35
  normal.qu, 6, 15, 23
  npp, 16, 21
  plot.iif, 10, 17
  plot.irf, 11, 17, 18, 30
  plot.tif, 19, 32
  plot.trf, 20, 33
  qrs, 13, 17, 21, 30
  read.ip.bilog, 21
  read.ip.icl, 22
  read.qu.icl, 6, 15, 23
  read.resp, 23
  sca, 15, 23, 24
  sco, 24, 26
  scp, 27
  sim, 28
  tet, 29
  tgp, 29
  tia, 30
  tif, 31
  trf, 32
  tsc, 33
  wle, 34
  api, 4
  dpv, 5
  eap, 5, 6, 13–15, 23, 28, 33, 35
  est, 7
  iif, 9, 17, 32
  irf, 10, 17, 18, 30, 33
  irtoys (irtoys-package), 2
  irtoys-package, 2
  itf, 11, 21
  mlebme, 5, 6, 14, 28, 33, 35
  normal.qu, 6, 15, 23
  npp, 16, 21
  plot.iif, 10, 17
  plot.irf, 11, 17, 18, 30
  plot.tif, 19, 32
  plot.trf, 20, 33
  qrs, 13, 17, 21, 30
  read.ip.bilog, 21
  read.ip.icl, 22
  read.qu.icl, 6, 15, 23
  read.resp, 23
  sca, 15, 23, 24
  sco, 24, 26
  SCP, 27
  Scored, 27
  Scored2pl, 27
  scp, 27
  sim, 28
  tet, 29
  tgp, 24, 29
  tia, 30
  tif, 19, 31
  trf, 20, 28, 32, 33
  tsc, 28, 33
  api, 4
INDEX

Unscored, 34
wle, 34