Package: shortIRT (via r-universe)

June 4, 2026

Type Package

Title Procedures Based on Item Response Theory Models for the Development of Short Test Forms

Version 2.0.0

Maintainer Ottavia M. Epifania <ottavia.epifania@unitn.it>

Description Implement different Item Response Theory (IRT) based procedures for the development of tests from item bank. The procedures are flexible enough to be adopted for the development of short forms of full-length tests. Different procedures are considered (Epifania, Anselmi & Robusto, 2022 <doi:10.1007/978-3-031-27781-8_7> and Epifania & Finos, 2025 <doi:10.1007/978-3-031-95995-0_32>). The main difference between the presented procedures refers to the degree of control that they allow for targeting specific latent trait levels. The simplest procedure, denoted as benchmark procedure, does not allow for any control on the latent trait levels of interest, while the other procedures allow for specifying either discrete latent trait levels for which the information needs to be maximized (theta-target procedure, <doi:10.1007/978-3-031-27781-8_7>) or a target information function that needs to be recreated with the selected items (item selection algorithm -ISA- denoted as Frank in <doi:10.1007/978-3-031-95995-0_32>). Another difference concerns the definition of the number of items to be selected. In the benchmark and theta-target procedures, the number of items must be defined a priori, while in ISA the number of items is determined automatically by the algorithm.

License MIT + file LICENSE

Encoding UTF-8

Imports ggplot2

RoxygenNote 7.3.3

Suggests MASS, rmarkdown, sirt, testthat (>= 3.0.0), V8

Config/testthat/edition 3

NeedsCompilation no

Author Ottavia M. Epifania [aut, cre] (ORCID: <https://orcid.org/0000-0001-8552-568X>), Pasquale Anselmi [ctb], Egidio Robusto [ctb], Livio Finos [ctb]

Repository https://cran.r-universe.dev

Date/Publication 2026-06-04 17:27:41 UTC

RemoteUrl https://github.com/cran/shortIRT

RemoteRef HEAD

RemoteSha cc37b1fd8512790bdf18f8bfc6687f7b5ac40981

Contents

---

bench *Benchmark Procedure*

Description

Develop a test or a short form of a test given the parameters of dichotomous or polytomous items in an item bank/full-length test according to the benchmark procedure. See `Details`.

Usage

```
bench(item_pars = NULL, iifs = NULL, theta = NULL, num_item = NULL, K = NULL)
```

Arguments

`item_pars` — `data.frame` with number of rows equal to the number of items. For dichotomous items, the dataframe must have 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location b_i, pseudo-guessing c_i, and upper asymptote e_i. For polytomous items, the dataframe has $2K$ columns, where K is the number of thresholds of the items (number of response categories -1). The first K columns correspond to step discrimination parameters $a_1, \ldots, a_K$ (must be named "a"), and the last K columns correspond to step difficulty (threshold) parameters $b_1, \ldots, b_K$ (must be named "b").

`iifs` — `data.frame` with number of rows equal to the length of the latent trait θ and number of columns equal to the number of items in the item bank. It contains the item information functions (IIFs) of the items in item bank/the full-length test. The arguments `item_pars` and `iifs` cannot be used together.

`theta` — `numeric` vector with the latent trait values.

`num_item` — `integer` defining the number N of items to include in the test.

`K` — `integer` defining the number of thresholds for the categories of the polytoumous items (i.e., number of response categories minus 1). Default is `NULL` (assumes dichotomous items).

Details

Let N be the number of items to be included in the test developed from an item bank B. The test $Q_{\text{bench}} \subseteq B$ with $|Q| = N$ is constructed by selecting the N items with the highest item information values, with no explicit reference to any specific level of the latent trait.

Given that $I_i(\theta)$ is the IIF for each item $i \in B$, the maximum value of its information function over θ is computed, as to define the vector

$$\mathbf{m} = (m_1, \ldots, m_{|B|}),$$

where

$$m_i = \max_{\theta} I_i(\theta), \qquad i = 1, \ldots, |B|.$$

The vector $\mathbf{m}$ is then sorted in decreasing order, and the first N items in the ordered vector (i.e., the items with the highest information functions), with $N \leq |B|$, are selected to form the test.

Further details on the benchmark procedure can be found in Epifania et al. (2022).

Value

An object of class `bench` of length 3 with:

- `test`: dataframe with the items selected for inclusion in the test (`isel`), their maximum information function (`maxiif`), for a specific latent trait level θ (column `theta`).
- `item_pars`: the original dataframe containing the item parameters.
- `selected_items`: dataframe with the parameters of the selected items.

- K: number of thresholds for the response categories of the items. If the items are dichotomous K is NULL.

References

Epifania, O. M., Anselmi, P., & Robusto, E. (2022). Item response theory approaches for test shortening. In M. Wiberg, D. Molenaar, J. Gonzalez, J. S. Kim, & H. Hwang (Eds.), Quantitative Psychology (Vol. 422, pp. 75–83). Springer Proceedings in Mathematics and Statistics. Springer, Cham. https://doi.org/10.1007/978-3-031-27781-8_7

Examples

```
# set a seed for the reproducibility of the results
set.seed(123)
# define the number of items in the item bank
n <- 50
# generate 500 random values of theta from a normal distribution with sd = 2
theta <- rnorm(500, sd = 2)
# generate item parameters  of the items in the item bank according to the 2-PL model
item_pars <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
# apply benchmark procedure
resB <- bench(item_pars, theta = theta, num_item = 5)
str(resB)
# generate an item bank with 4 polytomous items with K = 3
item_pars <- data.frame(matrix(c(
 1.2, 1.0, 0.8,  -1.0, 0.0, 1.2,
 0.9, 1.1, 1.3,  -0.5, 0.7, 1.8,
 0.5, 1.5, 1, -1.5, -1.0, 0,
 1, 1, 1, -1.5, -0, 0.5
 ), nrow = 4, byrow = TRUE))
# rename the columns
colnames(item_pars) = paste(rep(c("a", "b"), each = 3), 1:3, sep = "")
# apply benchmark procedure on polytomous items
resB_poly <- bench(item_pars, theta = theta, num_item = 2, K = 3)
str(resB_poly)
```

---

define_targets   *Define θ targets*

Description

Define θ targets either by considering the midpoints of equal intervals defined on the latent trait (equal) or the centroids obtained by clustering the latent trait (clusters). Further details on targets definition can be found in Epifania et al. (2022).

Usage

```
define_targets(theta, num_targets = NULL, method = c("equal", "clusters"))
```

Arguments

theta	numeric vector defining the latent trait θ.
num_targets	integer defining the number of θ targets. The number of θ targets defines the number of items included in the test.
method	character, either equal (default) or clusters.

Value

A vector of length num_targets with the generated θ targets. The class can be either equal or clusters, depending on the method used for the definition of the θ targets.

References

Epifania, O. M., Anselmi, P., & Robusto, E. (2022). Item response theory approaches for test shortening. In M. Wiberg, D. Molenaar, J. Gonzalez, J. S. Kim, & H. Hwang (Eds.), Quantitative Psychology (Vol. 422, pp. 75–83). Springer Proceedings in Mathematics and Statistics. Springer, Cham. https://doi.org/10.1007/978-3-031-27781-8_7

Examples

```
# set a seed for the reproducibility of the results
set.seed(123)
# generate 1000 random values of theta from a normal distribution
theta <- rnorm(1000)
# extract theta targets as the centroids of the clusters
targets <- define_targets(theta, num_targets = 5, method = "clusters")
```

---

i_info *Item Information Function (single item, $I\_i(\theta)$)*

Description

Compute the item information function $I_i(\theta)$ for a single dichotomous or polytoumous item under either the 4-PL model (dichotomous item) or the Generalized Partial Credit model (polytomous item). Specific models (e.g., 3-PL, 2-PL, 1-PL, or PCM, Rating Scale) are obtained by imposing constraints on the item parameters. See Details.

Usage

```
i_info(item_pars, theta = seq(-5, 5, length.out = 1000), K = NULL)
```

Arguments

item_pars
: `data.frame` with number of rows equal to the number of items. For dichotomous items, the dataframe must have 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location b_i, pseudo-guessing c_i, and upper asymptote e_i. For polytomous items, the dataframe has $2K$ columns, where K is the number of thresholds of the items (number of response categories -1). The first K columns correspond to step discrimination parameters $a_1, \ldots, a_K$ (must be named "a"), and the last K columns correspond to step difficulty (threshold) parameters

theta
: `numeric` vector of latent trait values. Default is a vector of a thousand values ranging from -5 to +5

K
: `integer` defining the number of thresholds for the categories of the polytoumous items (i.e., number of response categories minus 1). Default is `NULL` (assumes dichotomous items).

Details

Let $P(\theta)$ denote the probability of a correct response $x_{pi} = 1$ for person p (with latent trait level defined as θ_p) on item i under the four-parameter logistic (4-PL; Barton & Lord, 1981) model is defined as:

$$P(\theta) = c_i + \frac{e_i - c_i}{1 + \exp\left[-a_i(\theta_p - b_i)\right]}$$

where a_i is the discrimination parameter, b_i is the difficulty parameter (or location of item i on the latent trait), c_i is the lower asymptote (pseudo-guessing probability), and e_i is the upper asymptote (inattention/slip). By constraining $e_i = 1$, $c_i = 0$, and $a_i = 1 \; \forall i$, the probability is computed according to the 3-PL (Lord, 1980), 2-PL (Birnbaum, 1968) and 1-PL or the Rasch model (Rasch, 1960), respectively.

Let $Q(\theta) = 1 - P(\theta)$, the information function of item i is computed as:

$$I_i(\theta) = \frac{a_i^2 \left[P(\theta) - c_i\right]^2 \left[e_i - P(\theta)\right]^2}{(e_i - c_i)^2 \, P(\theta) \, Q(\theta)}$$

According to the Generalized Partial Credit Model (GPCM; Muraki, 1997), for a polytomous item with K thresholds separating the $K + 1$ categories, the probability of category k is defined as:

$$P(Y = k \mid \theta) = \frac{\exp\left(\sum_{k=1}^{K} a_k(\theta - b_k)\right)}{\sum_{j=0}^{K} \exp\left(\sum_{k=1}^{K} a_k(\theta - b_k)\right)}$$

where a_k and b_k are the discrimination and location parameters associate with each threshold k. If $a_k = 1$, $\forall k$, the Partial Credit Model (PCM, Muraki, 1992) is obtained.

The item information is computed as:

$$I_i(\theta) = \sum_{k=0}^{K} \frac{[P_k'(\theta)]^2}{P_k(\theta)}$$

Value

A numeric vector of length equal to `theta`, which contains the item information function for a single item with respect to the values specified in `theta`

References

Barton, M. A., & Lord, F. M. (1981). An upper asymptote for the three-parameter logistic item-response model. ETS Research Report Series, 1981(1), i–8. Princeton, NJ: Educational Testing Service.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord & M. R. Novick (Eds.), Statistical theories of mental test scores (pp. 397–479). Reading, MA: Addison-Wesley.

Lord, F. M. (1980). Applications of item response theory to practical testing problems. Hillsdale, NJ: Lawrence Erlbaum Associates.

Muraki, G. (1992). A generalized partial credit model: Application of an EM algorithm. Psychometrika, 57(2), 159–176.

Muraki, G. (1997). A generalized partial credit model with step discrimination. Journal of Educational Measurement, 34(2), 115–127.

Rasch, G. (1960). Probabilistic models for some intelligence and attainment tests. Copenhagen, Denmark: Danish Institute for Educational Research.

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Create a sequence of latent trait values
# spanning the ability continuum
theta <- seq(-4, 4, length.out = 200)

# Define parameters for a dichotomous item
# b = difficulty
# a = discrimination
# c = lower asymptote (guessing)
# e = upper asymptote
item_par <- data.frame(
  b = 0,
  a = 1.5,
  c = .10,
  e = .98
)

# Compute item information function (IIF)
# across the theta continuum
info_dichotomous <- i_info(item_par, theta = theta)

# Define parameters for a 4-category item
# (K = 3 thresholds / category transitions)
# a's = category discrimination parameters
```

```
# b's = threshold/location parameters
item_pars <- data.frame(
  a1 = 1.2,
  a2 = 1.0,
  a3 = 0.8,
  b1 = -1.0,
  b2 = 0.0,
  b3 = 1.2
)

# Compute item information for the polytomous item
# across the theta continuum
info <- i_info(item_pars, theta = theta, K = 3)
```

IRT *Compute expected probability for a single dichotomous item*

Description

Compute the expected probability for an item i given the latent trait θ and the item parameters. Depending on the parameters that are specified, the probability is computed according to the 1-PL, 2-PL, 3-PL, or 4-PL models.

Usage

```
IRT(theta, b = 0, a = 1, c = 0, e = 1)
```

Arguments

`theta`	`numeric` defining the latent trait level of person p. It can be a single value or a vector of values.
`b`	`numeric` defining the location of item i. Default is 0.
`a`	`numeric` defining the discrimination parameter of item i. Default is 1.
`c`	`numeric` defining the lower asymptote (pseudo-guessing parameter) of item i. Default is 0.
`e`	`numeric` defining the upper asymptote (inattention) of item i. Default is 1.

Details

The probability of a correct response $x_{pi} = 1$ for person p (with latent trait level defined as θ_p) on item i under the four-parameter logistic (4-PL; Barton & Lord, 1981) model is defined as:

$$P(x_{pi} = 1 \mid \theta_p, b_i, a_i, c_i, e_i) = c_i + \frac{e_i - c_i}{1 + \exp\left[-a_i(\theta_p - b_i)\right]}$$

where a_i is the discrimination parameter, b_i is the difficulty parameter (or location of item i on the latent trait), c_i is the lower asymptote (pseudo-guessing probability), and e_i is the upper asymptote (inattention/slip). By constraining $e_i = 1$, $c_i = 0$, and $a_i = 1\ \forall i$, the probability is computed according to the 3-PL (Lord, 1980), 2-PL (Birnbaum, 1968) and 1-PL or the Rasch model (Rasch, 1960), respectively.

Value

a single value, that is the probability of the correct response for item i given the specified parameters

References

Barton, M. A., & Lord, F. M. (1981). An upper asymptote for the three-parameter logistic item-response model. ETS Research Report Series, 1981(1), i–8. Princeton, NJ: Educational Testing Service.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord & M. R. Novick (Eds.), Statistical theories of mental test scores (pp. 397–479). Reading, MA: Addison-Wesley.

Lord, F. M. (1980). Applications of item response theory to practical testing problems. Hillsdale, NJ: Lawrence Erlbaum Associates.

Rasch, G. (1960). Probabilistic models for some intelligence and attainment tests. Copenhagen, Denmark: Danish Institute for Educational Research.

Examples

```
# compute the probability for an item according to the 1-PL model
IRT(theta = 0, b = 0,  a = 1, c = 0, e = 1)
# compute the probability for a vector of thetas for the same item
IRT(theta = c(-1, 0, 1), b = 0,  a = 1, c = 0, e = 1)
# compute the probability for a vector of thetas for an item according to the 4-PL model
IRT(theta = c(-1, 0, 1), b = 0,  a = 1.25, c = 0.10, e = 0.98)
```

---

irt_estimate — *Estimate of θ via Maximum Likelihood*

Description

Maximum Likelihood estimation of θ

Usage

```
irt_estimate(item_par, responses = NULL, theta, lower = -3, upper = abs(lower))
```

Arguments

`item_par`	`data.frame`, dataframe with nrows equal to the number of items and 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location b_i, pseudo-guessing c_i, and upper asymptote e_i.
`responses`	`matrix`, $P \times I$ matrix with the dichotomous responses of each respondent p on each item i. Default is `NULL`.
`theta`	`numeric` vector with true values of θ
`lower`	`integer` lower value of θ to be considered for the estimation
`upper`	`integer` upper value of θ to be considered for the estimation

Value

A numeric vector of length equal to the length of `theta` with the ML estimation of the latent trait

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500)
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
# estimate theta
theta_hat <- irt_estimate(item_par, theta = theta)
plot(theta, theta_hat)
```

`isa` *Item Selection Algorithm*

Description

Develop a test or a short form given the parameters of dichotomous or polytomous in an item bank/full-length test according to the Item Selection Algorithm (ISA, Epifania & Finos, 2025). See `Details`.

Usage

```
isa(item_pars, tif_target, nmin = round(nrow(item_pars) * 0.1), K = NULL)
```

Arguments

`item_pars` — `data.frame` with number of rows equal to the number of items. For dichotomous items, the dataframe must have 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location b_i, pseudo-guessing c_i, and upper asymptote e_i. For polytomous items, the dataframe has $2K$ columns, where K is the number of thresholds of the items (number of response categorie -1). The first K columns correspond to step discrimination parameters $a_1, \ldots, a_K$ (must be named "a"), and the last K columns correspond to step difficulty (threshold) parameters $b_1, \ldots, b_K$ (must be named "b").

`tif_target` — `data.frame` with two columns: (i) `theta` the latent trait θ and (2) `tif` defining the values of the TIF target. The TIF target should be computed as the mean TIF to allow for the comparability with the TIF obtained from the test.

`nmin` — `integer` defining the minimum number of items to be included in the test (i.e., the termination criterion is not tested until the minimum number of items is reached). Default is the 10% of the total number of items.

K `integer` defining the number of thresholds for polytomous items (number of response categories minus 1). Default is `NULL` (assumes dichotomous items).

Details

Let $t = 0, \ldots, T$ denote the iteration index of the procedure, TIF' denote the test information target, and TIF^t denote the test information function obtained from $Q^t_{\text{isa}} \subset B$ (where B is the item bank and Q^t is the subset of items selected up to iteration t). At $t = 0$: $\text{TIF}^0(\theta) = 0, \forall\theta$, $Q^0 = \emptyset$. For $t \geq 0$,

1. Consider the available items at iteration t
$$A^t = B \setminus Q^t$$
2. Compute the provisional TIF (pTIF_i) considering the available items one at the time
$$\forall i \in A^t, \text{pTIF}_i := \frac{\text{TIF}^t + I_i(\theta)}{|Q^t| + 1}$$
3. Select a provisional item i^* allowing for minimizing the distance from the TIF target
$$i^* := \arg\min_{i \in A^t} \text{abs}(\text{TIF}^* - \text{pTIF}_i)$$
4. Test the termination criterion: If
$\text{abs}(\text{TIF}^* - \text{pTIF}_{i^*}) \ngeq \text{abs}(\text{TIF}^* - \text{TIF}^t), Q^{t+1} = Q^t \cup \{i^+\}$, $\text{TIF}^{t+1} = \text{pTIF}_{i^+}$, Go back to 1
5. Stop, $Q_{\text{ISA}} = Q^t$

Further details on the algorithm can be found in Epifania & Finos (2025), where the algorithm is denoted as Frank.

Value

An object of class `isa` of length 6 containing:

- **test**: a dataframe containing the items selected for inclusion in the test (column `isel`) and the minimum number of items set in `nmin`.
- **item_pars**: the original dataframe containing the item parameters.
- **selected_items**: a dataframe containing the parameters of the selected items.
- **all_iifs**: a dataframe containing the IIFs of all the original items.
- **tif_target**: the dataframe with the TIF target used for the item selection.
- **K**: Number of thresholds for the response categories of the items. If the items are dichotomous `K` is `NULL`.

References

Epifania, O. M., & Finos, L. (2025). Nothing lasts forever – only item administration: An item response theory algorithm to shorten tests. In E. Di Bella, V. Gioia, C. Lagazio, & S. Zaccarin (Eds.), Statistics for Innovation III (pp. 188–193). Italian Statistical Society Series on Advances in Statistics. Springer, Cham. https://doi.org/10.1007/978-3-031-95995-0_32

Examples

```
# set a seed for the reproducibility of the results
set.seed(123)
# define the number of items in the item bank
n <- 50
# generate 500 random values of theta from a normal distribution with sd = 2
theta <- rnorm(500, sd = 2)
# generate item parameters  of the items in the item bank according to the 2-PL model
item_pars <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
# define the tif target as the average tif of the items in the item bank
tif_target <- tif(
  item_info(item_pars),
  fun = "mean"
)
# apply ISA with the constraint of selecting at least 4 items
resIsa <- isa(item_pars, tif_target, nmin = 4)
str(resIsa)

# generate an item bank with 4 polytomous items with K = 3
item_pars <- data.frame(
  matrix(c(
    1.2, 1.0, 0.8,  -1.0,  0.0,  1.2,
    0.9, 1.1, 1.3,  -0.5,  0.7,  1.8,
    0.5, 1.5, 1.0,  -1.5, -1.0,  0.0,
    1.0, 1.0, 1.0,  -1.5,  0.0,  0.5
  ), nrow = 4, byrow = TRUE)
)

colnames(item_pars) <- paste(
  rep(c("a", "b"), each = 3),
  1:3,
  sep = ""
)
# rename the columns
# apply ISA with the constraint of selecting at least 2 items
resIsa_poly <- isa(item_pars, tif_target, nmin = 2, K = 3)
str(resIsa_poly)
```

`item_info` *Item Information Functions (multiple items, $I\_i(\theta)$)*

Description

Computes the item information functions for multiple dichotomous or polytomous items

Usage

```
item_info(item_pars, theta = seq(-5, 5, length.out = 1000), K = NULL)
```

Arguments

`item_pars`	`data.frame` with number of rows equal to the number of items. For dichotomous items, the dataframe must have 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location b_i, pseudo-guessing c_i, and upper asymptote e_i. For polytomous items, the dataframe has $2K$ columns, where K is the number of thresholds of the items (number of response categories -1). The first K columns correspond to step discrimination parameters $a_1, \ldots, a_K$ (must be named "a"), and the last K columns correspond to step difficulty (threshold) parameters
`theta`	`numeric` vector of latent trait values. Default is a vector of A thousand values ranging from -5 to +5
`K`	`integer` defining the number of thresholds for the categories of the polytoumous items (i.e., number of response categories minus 1). Default is `NULL` (assumes dichotomous items).

Details

Let $P(\theta)$ denote the probability of a correct response $x_{pi} = 1$ for person p (with latent trait level defined as θ_p) on item i under the four-parameter logistic (4-PL; Barton & Lord, 1981) model is defined as:

$$P(\theta) = c_i + \frac{e_i - c_i}{1 + \exp\left[-a_i(\theta_p - b_i)\right]}$$

where a_i is the discrimination parameter, b_i is the difficulty parameter (or location of item i on the latent trait), c_i is the lower asymptote (pseudo-guessing probability), and e_i is the upper asymptote (inattention/slip). By constraining $e_i = 1$, $c_i = 0$, and $a_i = 1\ \forall i$, the probability is computed according to the 3-PL (Lord, 1980), 2-PL (Birnbaum, 1968) and 1-PL or the Rasch model (Rasch, 1960), respectively.

Let $Q(\theta) = 1 - P(\theta)$, the information function of item i is computed as:

$$I_i(\theta) = \frac{a_i^2 \left[P(\theta) - c_i\right]^2 \left[e_i - P(\theta)\right]^2}{(e_i - c_i)^2\, P(\theta)\, Q(\theta)}$$

According to the Generalized Partial Credit Model (GPCM; Muraki, 1997), for a polytomous item with K thresholds separating the $K + 1$ categories, the probability of category k is defined as:

$$P(Y = k \mid \theta) = \frac{\exp\left(\sum_{k=1}^{K} a_k(\theta - b_k)\right)}{\sum_{j=0}^{K} \exp\left(\sum_{k=1}^{K} a_k(\theta - b_k)\right)}$$

where a_k and b_k are the discrimination and location parameters associate with each threshold k. If $a_k = 1$, $\forall k$, the Partial Credit Model (PCM, Muraki, 1992) is obtained.

The item information is computed as:

$$I_i(\theta) = \sum_{k=0}^{K} \frac{[P_k'(\theta)]^2}{P_k(\theta)}$$

Value

A `matrix` of class `iifs` with number of rows equal to the length of `theta` and number of columns equal to the number of items in `item_par`

References

Barton, M. A., & Lord, F. M. (1981). An upper asymptote for the three-parameter logistic item-response model. ETS Research Report Series, 1981(1), i–8. Princeton, NJ: Educational Testing Service.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord & M. R. Novick (Eds.), Statistical theories of mental test scores (pp. 397–479). Reading, MA: Addison-Wesley.

Lord, F. M. (1980). Applications of item response theory to practical testing problems. Hillsdale, NJ: Lawrence Erlbaum Associates.

Muraki, G. (1992). A generalized partial credit model: Application of an EM algorithm. Psychometrika, 57(2), 159–176.

Muraki, G. (1997). A generalized partial credit model with step discrimination. Journal of Educational Measurement, 34(2), 115–127.

Rasch, G. (1960). Probabilistic models for some intelligence and attainment tests. Copenhagen, Denmark: Danish Institute for Educational Research.

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Define parameters for five dichotomous items
# b = difficulty parameters
# a = discrimination parameters
# c = lower asymptote
# e = upper asymptote
parameters <- data.frame(
  b = c(-3, -2, 0, 2, 3),
  a = runif(5, 1.2, 1.9),
  c = rep(0, 5),
  e = rep(1, 5)
)

# Compute item information functions for
# dichotomous items using default theta values
infos <- item_info(parameters)

# Display the first rows of the information matrix
```

```
head(infos)

# Define parameters for four polytomous items
# with four response categories (K = 3 thresholds)
item_pars <- data.frame(
  matrix(
    c(
      1.2, 1.0, 0.8,  -1.0,  0.0, 1.2,
      0.9, 1.1, 1.3,  -0.5,  0.7, 1.8,
      0.5, 1.5, 1.0,  -1.5, -1.0, 0.0,
      1.0, 1.0, 1.0,  -1.5,  0.0, 0.5
    ),
    nrow = 4,
    byrow = TRUE
  )
)

# Assign parameter names:
# a1-a3 = discrimination parameters
# b1-b3 = threshold/location parameters
colnames(item_pars) <- paste(
  rep(c("a", "b"), each = 3),
  1:3,
  sep = ""
)

# Compute item information functions
# for the polytomous items
info_poly <- item_info(item_pars, K = 3)

# Display the first rows of the information matrix
head(info_poly)
```

`logLik_theta` *Log-likelihood estimation of* θ

Description

Provide the log-likelihood function for estimating θ given the item parameters (dichotomous only) and the true values of θ

Usage

```
logLik_theta(theta, x, item_par)
```

Arguments

`theta`	`numeric` vector with true values of θ
`x`	`integer` vector of 0s and 1s, response pattern of each respondent

item_par — `data.frame`, dataframe with nrows equal to the number of items and 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location b_i, pseudo-guessing c_i, and upper asymptote e_i.

Value

The function for estimating the log-likelihood of θ

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500)
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
obs_response <- obsirt(mpirt(item_par, theta))
# LogLikelihood of theta
logLik_theta(theta, obs_response, item_par)
```

---

mpirt — *Compute expected probability for multiple dichotomous items*

Description

Compute the expected probability for multiple dichotomous items given the latent trait levels θ and the item parameters. Depending on the parameters that are specified, the probability is computed according to the 1-PL, 2-PL, 3-PL, or 4-PL models.

Usage

```
mpirt(item_pars, theta)
```

Arguments

item_pars — `data.frame` with number of rows equal to the number of items and 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location/difficulty b_i, pseudo-guessing c_i, and upper asymptote e_i.

theta — `numeric` defining the latent trait level of person p, it can be a single value or a vector of values.

Details

The probability of a correct response $x_{pi} = 1$ for person p (with latent trait level defined as θ_p) on item i under the four-parameter logistic (4-PL; Barton & Lord, 1981) model is defined as:

$$P(x_{pi} = 1 \mid \theta_p, b_i, a_i, c_i, e_i) = c_i + \frac{e_i - c_i}{1 + \exp\left[-a_i(\theta_p - b_i)\right]}$$

where a_i is the discrimination parameter, b_i is the difficulty parameter (or location of item i on the latent trait), c_i is the lower asymptote (pseudo-guessing probability), and e_i is the upper asymptote (inattention/slip). By constraining $e_i = 1$, $c_i = 0$, and $a_i = 1 \; \forall i$, the probability is computed according to the 3-PL (Lord, 1980), 2-PL (Birnbaum, 1968) and 1-PL or the Rasch model (Rasch, 1960), respectively.

Value

A $P \times I$ (where P is the number of respondents corresponding to the length of `theta` and I is the number of items corresponding to the number of rows in `item_pars`) matrix of class `mpirt` with the expected probability of observing a correct response for respondent p on item i

References

Barton, M. A., & Lord, F. M. (1981). An upper asymptote for the three-parameter logistic item-response model. ETS Research Report Series, 1981(1), i–8. Princeton, NJ: Educational Testing Service.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord & M. R. Novick (Eds.), Statistical theories of mental test scores (pp. 397–479). Reading, MA: Addison-Wesley.

Lord, F. M. (1980). Applications of item response theory to practical testing problems. Hillsdale, NJ: Lawrence Erlbaum Associates.

Rasch, G. (1960). Probabilistic models for some intelligence and attainment tests. Copenhagen, Denmark: Danish Institute for Educational Research.

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Define the number of items
n <- 50

# Generate latent trait values (theta) for 500 respondents
theta <- rnorm(500)

# Create item parameter matrix/data frame
# b = item difficulty parameters
# a = item discrimination parameters
# c = lower asymptote (guessing parameter)
# e = upper asymptote
item_pars <- data.frame(
```

```
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)

# Compute expected response probabilities
# for each respondent-item combination
expected_prob <- mpirt(item_pars, theta)
```

`obsirt` *Simulate dichotomous responses according to IRT probabilities*

Description

Simulate dichotomous responses according to IRT probabilities simulated with the `mpirt()` function .

Usage

```
obsirt(myp)
```

Arguments

`myp`	Object of class `mpirt` containing the expected IRT probabilities obtained with function `mpirt()`

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Define the number of items
n <- 50

# Generate latent trait values (theta) for 500 respondents
theta <- rnorm(500)

# Create item parameter matrix/data frame
# b = item difficulty parameters
# a = item discrimination parameters
# c = lower asymptote (guessing parameter)
# e = upper asymptote
item_pars <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
```

```
# Compute expected response probabilities
# for each respondent-item combination
expected_prob <- mpirt(item_pars, theta)

# Generate observed item responses from the
# expected probabilities
simulated_responses <- obsirt(expected_prob)
```

---

plot.bench *Method for plotting the TIF of the test/short test form*

Description

The test/short test form is obtained with the benchmark procedure implemented with function bench(). Details on the procedure can be found in the documentation of the bench() function.

Usage

```
## S3 method for class 'bench'
plot(x, fun = "sum", ...)
```

Arguments

x	Object of class bench
fun	character, whether to consider the mean or the sum for the computation of the TIF
...	other arguments

Value

A ggplot showing the TIFs of the test.

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Define the number of items
n <- 50

# Generate latent trait values for 500 respondents
# using a wider latent distribution (sd = 2)
theta <- rnorm(500, sd = 2)

# Create item parameter matrix/data frame
# b = difficulty parameters
# a = discrimination parameters
# c = lower asymptote
# e = upper asymptote
```

```
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)

# Run benchmark/selection procedure
# selecting 5 items from the item pool
resB <- bench(
  item_par,
  theta = theta,
  num_item = 5
)

# Plot benchmark results including
# item-level and test-level information
plot(resB)

# Plot only the Test Information Function (TIF)
plot(resB, show_both = FALSE)

# Define parameters for four polytomous items
# with four response categories (K = 3 thresholds)
item_pars <- data.frame(
  matrix(
    c(
      1.2, 1.0, 0.8,  -1.0,  0.0, 1.2,
      0.9, 1.1, 1.3,  -0.5,  0.7, 1.8,
      0.5, 1.5, 1.0,  -1.5, -1.0, 0.0,
      1.0, 1.0, 1.0,  -1.5,  0.0, 0.5
    ),
    nrow = 4,
    byrow = TRUE
  )
)

# Assign parameter names
colnames(item_pars) <- paste(
  rep(c("a", "b"), each = 3),
  1:3,
  sep = ""
)

# Run benchmark/selection procedure
# for polytomous items selecting 2 items
resB_poly <- bench(
  item_pars,
  theta = theta,
  num_item = 2,
  K = 3
)
```

```
# Plot benchmark results for the polytomous case
plot(resB_poly)
```

---

plot.iifs *Method for plotting the item information functions*

Description

Plot the information functions of polytomous or dichotomous items

Usage

```
## S3 method for class 'iifs'
plot(x, single_panels = TRUE, items = NULL, ...)
```

Arguments

x	data.frame of class iifs obtained with the function item_info()
single_panels	logical, default is TRUE. Whether to show the $I_i(\theta)$ of each item on a different panel
items	default is NULL (shows all items). Allows for selecting specific items for the plot.
...	other arguments

Details

If more there are more than 10 items, the legend associated to the color of the lines is not displayed.

Value

A ggplot

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Simulate parameters for five dichotomous items
# according to a 2-PL specification
# b = difficulty parameters
# a = discrimination parameters
# c = lower asymptote
# e = upper asymptote
parameters <- data.frame(
  b = c(-3, -2, 0, 2, 3),
  a = runif(5, 1.2, 1.9),
  c = rep(0, 5),
  e = rep(1, 5)
)
```

```
# Compute item information functions (IIFs)
infos <- item_info(parameters)

# Plot information functions for all items
plot(infos)

# Plot information functions only for items 1 and 3
# on a single panel
plot(
  infos,
  items = c(1, 3),
  single_panels = FALSE
)
```

---

plot.isa *Method for plotting the TIF of the test/short test form*

Description

The test/short test form is obtained with the ISA procedure implemented with function `isa()`. Details on the procedure can be found in the documentation of the `isa()` function.

Usage

```
## S3 method for class 'isa'
plot(x, fun = "mean", show_all = FALSE, show_dist = FALSE, ...)
```

Arguments

x	Object of class `isa` obtained with function `isa()`
fun	`character`, whether to consider the mean or the sum for the computation of the TIF
show_all	`logical`, default is `FALSE`. Whether to show the TIF of the test and the TIF target together with the TIF obtained from all the items in `item_par`
show_dist	`logical`, default is `FALSE`. Whether to show or not the difference and distance (absolute difference) from the TIF target.
...	other arguments

Value

A `ggplot` showing either the TIFs of the test, that of the item bank, and the TIF target or the distance/difference.

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Define the number of items in the item bank
n <- 50

# Generate latent trait values (not directly used here,
# but typically included in IRT simulations)
theta <- rnorm(500)

# Create item parameter matrix/data frame
# b = difficulty parameters
# a = discrimination parameters
# c = lower asymptote
# e = upper asymptote
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)

# Compute item information functions for the item bank
# and define a target test information function (TIF)
# as the mean information across items
target <- tif(item_info(item_par), fun = "mean")

# Run item selection algorithm (ISA)
# selecting a minimum of 5 items
resI <- isa(item_par, target, nmin = 5)

# Plot selected item set and related results
plot(resI)

# Show the Test Information Function (TIF)
# of the selected item bank
plot(resI, show_all = TRUE)

# Show the distance between obtained TIF and target TIF
plot(resI, show_dist = TRUE)
```

`plot.theta_target` *Method for plotting the TIF of the test/short test form*

Description

The test/short test form is obtained with the theta target procedure implemented with function `theta_target()`. Details on the procedure can be found in the documentation of the `theta_target()` function.

Usage

```
## S3 method for class 'theta_target'
plot(x, fun = "sum", show_targets = TRUE, ...)
```

Arguments

x	Object of class `theta_target` obtained with function `theta_target()`
fun	`character`, whether to consider the mean or the sum for the computation of the TIF
show_targets	`logical`, default is `TRUE`. Whether to show or not the theta targets. If `TRUE` the theta targets are shown. The color associated to each theta target represents the specific item that has been selected for maximizing the information for that specific point.
...	other arguments

Details

If more than 10 theta targets are selected, the legend associated to each theta target is not displayed.

Value

A `ggplot` showing the TIFs of the test, the locations of the theta targets, and the items that satisfy each theta target.

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Define the number of items in the item bank
n <- 50

# Generate latent trait values for 500 respondents
theta <- rnorm(500)

# Create item parameter matrix/data frame
# b = difficulty parameters
# a = discrimination parameters
# c = lower asymptote
# e = upper asymptote
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)

# Define a theta targets
targets <- define_targets(theta, num_targets = 4)
```

```
# Perform theta-targeted item selection
resT <- theta_target(targets, item_par)

# Plot results of the target-based selection
plot(resT)

# Plot results without displaying theta target
plot(resT, show_targets = FALSE)
```

---

plot.tif *Plot TIF*

Description

Plot the test information function computed with the `tif()` function.

Usage

```
## S3 method for class 'tif'
plot(x, ...)
```

Arguments

`x`	object of class `tif` obtained with the `tif()` function
`...`	other arguments

Value

A `ggplot` displaying the TIF

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Define the number of items in the item bank
n <- 5

# Create item parameter matrix/data frame
# b = difficulty parameters
# a = discrimination parameters
# c = lower asymptote
# e = upper asymptote
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
```

```
# Compute item information functions (IIFs)
iifs <- item_info(item_par)

# Compute Test Information Function (TIF)
test_tif <- tif(iifs)

# Plot the test information function
plot(test_tif)

# Compute the mean TIF across items/components
test_tif_mean <- tif(iifs, fun = "mean")

# Plot the mean test information function
plot(test_tif_mean)
```

`summary.bench` *Method for the summary of the test/short test form*

Description

The test/short test form is obtained with the benchmark procedure implemented with function `bench()`. Details on the procedure can be found in the documentation of the `bench()` function.

Usage

```
## S3 method for class 'bench'
summary(object, ...)
```

Arguments

`object`	Object of class `bench()`
`...`	other arguments

Value

A summary of the test obtained from the application of the benchmark procedure

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Define the number of items in the item bank
n <- 50

# Generate latent trait values for 500 respondents
theta <- rnorm(500)
```

```
# Create item parameter matrix/data frame
# b = difficulty parameters
# a = discrimination parameters
# c = lower asymptote
# e = upper asymptote
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)

# Run benchmark/item selection procedure
# selecting 5 items from the item bank
resB <- bench(
  item_par,
  theta = theta,
  num_item = 5
)

# Summarize benchmark results
summary(resB)
```

`summary.isa` *Method for the summary of the test/short test form*

Description

The test/short test form is obtained with the ISA procedure implemented with function `isa()`. Details on the procedure can be found in the documentation of the `isa()` function.

Usage

```
## S3 method for class 'isa'
summary(object, ...)
```

Arguments

`object`	Object of class `isa`
`...`	other arguments

Value

A summary of the test obtained from the application of ISA

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Define the number of items in the item bank
n <- 50

# Create item parameter matrix/data frame
# b = difficulty parameters
# a = discrimination parameters
# c = lower asymptote
# e = upper asymptote
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)

# Compute item information functions and define
# a target Test Information Function (TIF)
# using the mean information across items
target <- tif(
  item_info(item_par),
  fun = "mean"
)

# Run item selection algorithm (ISA)
# selecting at least 5 items
resI <- isa(
  item_par,
  target,
  nmin = 5
)

# Summarize item selection results
summary(resI)
```

`summary.theta_target` *Method for the summary of the test/short test form*

Description

The test/short test form is obtained with the theta target procedure implemented with function `theta_target()`. Details on the procedure can be found in the documentation of the `theta_target()` function.

Usage

```
## S3 method for class 'theta_target'
summary(object, ...)
```

Arguments

`object`	Object of class `theta_target`
`...`	other arguments

Value

A summary of the test obtained from the application of the theta target procedure

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Define the number of items in the item bank
n <- 50

# Generate latent trait values for 500 respondents
theta <- rnorm(500)

# Create item parameter matrix/data frame
# b = difficulty parameters
# a = discrimination parameters
# c = lower asymptote
# e = upper asymptote
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)

# Define theta-based target regions
# splitting the latent trait distribution into 4 segments
targets <- define_targets(theta, num_targets = 4)

# Perform theta-targeted item selection
# matching item information to target regions
resT <- theta_target(targets, item_par)

# Summarize theta-targeted selection results
summary(resT)
```

`theta_target` *Theta target procedure*

Description

Develop a test or a short form given the parameters of dichotomous or polytomous in an item bank/full-length test according to the theta target procedure. See `Details`.

Usage

```
theta_target(
  targets,
  item_pars,
  theta = seq(-5, 5, length.out = 1000),
  K = NULL
)
```

Arguments

- `targets` — `numeric`, either a vector with the discrete values of theta for which the information needs to be maximized obtained with the `define_targets()` function or a vector with user-defined values. If the same theta value is defined and repeated several time, it can be passed as a named list, where `value` indicates the value that needs to be repeated and `num_targets` defines the number of times it is repeated for.
- `item_pars` — `data.frame` with nrows equal to the number of items. For dichotomous items, the matrix must have 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location b_i, pseudo-guessing c_i, and upper asymptote e_i. For polytomous items, the matrix has $2K$ columns, where K is the number of thresholds of the items (number of response categorie -1). The first K columns correspond to step discrimination parameters $a_1, \ldots, a_K$ (must be named "a"), and the last K columns correspond to step difficulty (threshold) parameters (must be named "b") $b_1, \ldots, b_K$.
- `theta` — `numeric` vector with the values of the latent trait θ (needed for the computation of the IIFs of all items)
- `K` — `integer` defining the number of thresholds for the categories of the polytoumous items (i.e., number of categories minus one). Default is `NULL` (assumes dichotomous items).

Details

Let Θ be the set of N θ-targets (θ'), i.e., the latent trait levels of interest for the assessment defined as discrete levels of θ, where N denotes the desired length of $Q_{\text{target}} \subseteq B$, where B denotes the item bank. The test $Q_{\text{target}} \subseteq B$ of length N is developed by considering the information $I_i(\theta'_n)$ that each item in B provides with respect to each $\theta' \in \Theta$. Therefore, an optimal item (i.e., the item with the highest information function) is chosen for each θ'. Given that N is the pre-defined length of test Q and that an optimal item is selected for each $\theta' \in \Theta$, then $|\Theta| = N$.

Let $t = 0, \ldots, T$ denote the iteration index of the procedure and define:

- $Q^t \subseteq B$ as the set of items selected for inclusion in the test up to iteration t;
- $S^t \subseteq \Theta$ as the set of θ's satisfied up to iteration t.

At $t = 0$, $Q^0 = \emptyset$ and $S^0 = \emptyset$.

The procedure iterates the following steps until $t = T$:

1. Select the item–target pair (i, n) maximizing the item information function:

$$(i, n) = \arg \max_{i \in B \setminus S^t,\ n \in N \setminus Q^t} \text{IIF}(i, n)$$

2. Update the set of selected items:

$$Q^{t+1} = Q^t \cup \{i\}$$

3. Update the set of satisfied ability targets:

$$S^{t+1} = S^t \cup \{n\}$$

At iteration T, the procedure yields $|Q^{T+1}| = N$ and $|S^{T+1}| = N$. Further details can be found in Epifania et al. (2022).

Value

An object of class `theta_target` of length 4 containing:

- **test**: a data frame containing the items selected for inclusion in the test (column `isel`), their maximum item information function (column `maxiif`), and the corresponding theta target (column `theta_target`).
- **item_pars**: the original data frame containing the item parameters.
- **selected_items**: a data frame containing the parameters of the selected items.
- **intervals**: a character string indicating how the theta targets were obtained. The value `"clusters"` denotes clustering of the latent trait, `"equal"` denotes equally spaced intervals, and `"unknown"` identifies any other case (e.g., user-defined theta targets).
- `K`: Number of thresholds for the response categories of the items. If the items are dichotomous `K` is `NULL`.

References

Epifania, O. M., Anselmi, P., & Robusto, E. (2022). Item response theory approaches for test shortening. In M. Wiberg, D. Molenaar, J. Gonzalez, J. S. Kim, & H. Hwang (Eds.), Quantitative Psychology (Vol. 422, pp. 75–83). Springer Proceedings in Mathematics and Statistics. Springer, Cham. https://doi.org/10.1007/978-3-031-27781-8_7

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500)
item_pars <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
targets <- define_targets(theta, num_targets = 4)
resT <- theta_target(targets, item_pars)
str(resT)
```

```
# polytomous items with user defined theta targets
item_pars <- data.frame(matrix(c(
 1.2, 1.0, 0.8,  -1.0, 0.0, 1.2,
 0.9, 1.1, 1.3,  -0.5, 0.7, 1.8,
 0.5, 1.5, 1, -1.5, -1.0, 0,
 1, 1, 1, -1.5, -0, 0.5
 ), nrow = 4, byrow = TRUE))
colnames(item_pars) = paste(rep(c("a", "b"), each = 3), 1:3, sep = "")
resT_poly <- theta_target(c(-1,0), item_pars, K = 3)
str(resT_poly)
```

`tif` *Test Information Function (TIF)*

Description

Compute the test information function of a test given a matrix of item information functions computed with the `item_info()` function. See `Details`.

Usage

```
tif(iifs, fun = "sum")
```

Arguments

`iifs`	object of class `iifs` containing the item information functions
`fun`	`character` defining the function for the computation of the TIF, either by summing the items (sum) or by computing the mean (mean)

Details

The test infromation function (TIF) for both polytomous and dichotomous items is computed as:

$$\text{TIF}(\theta) = \sum_{i=1}^{B} I_i(\theta)$$

Where B is the item bank.

Value

A `data.frame` of class `tif` with two columns: (i) `theta` containing the latent trait values, and (ii) `tif` containing the TIF values computed as either the sum or the mean of the IIFs

Examples

```
# Set random seed for reproducibility
set.seed(123)

# Generate latent trait values for 100 respondents
theta <- rnorm(100)

# Define the number of items
n <- 5

# Create item parameter matrix/data frame
# b = difficulty parameters
# a = discrimination parameters
# c = lower asymptote
# e = upper asymptote
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)

# Compute item information functions (IIFs)
# using default theta values
iifs <- item_info(item_par)

# Compute the test information function (TIF)
# by combining information across items
test_tif <- tif(iifs)
```

Index