Package ‘dscore’

July 22, 2025

Type Package

Title D-Score for Child Development

Version 1.10.0

Description The D-score summarizes the child's performance on a set of milestones into a single number. The package implements four Rasch model keys to convert milestone scores into a D-score. It provides tools to calculate the D-score and its precision from the child's milestone scores, to convert the D-score into the Development-for-Age Z-score (DAZ) using age-conditional references, and to map milestone names into a generic 9-position item naming convention.

Depends R (>= 4.1.0)

Imports dplyr (>= 1.0.0), Rcpp, stats, stringi, tidyr (>= 1.0.0)

LinkingTo Rcpp, RcppArmadillo

Suggests ggplot2, kableExtra, knitr, lme4, patchwork, rmarkdown, testthat

Encoding UTF-8

License AGPL-3

LazyData TRUE

VignetteBuilder knitr

NeedsCompilation yes

URL https://github.com/d-score/dscore, https://d-score.org/dscore/, https://d-score.org/dbook1/

BugReports https://github.com/d-score/dscore/issues

Copyright Stef van Buuren, Iris Eekhout, Arjan Huizing

RoxygenNote 7.3.2

Author Stef van Buuren [cre, aut],
Iris Eekhout [aut],
Arjan Huizing [aut],
Jonathan Seiden [aut]

Maintainer Stef van Buuren <stef.vanbuuren@tno.nl>

Repository CRAN

Date/Publication 2025-06-05 05:50:02 UTC

Contents

---

builtin_itembank *Collection of items fitting the Rasch model*

Description

A data frame with administrative information per item with difficulty estimates (tau) from the Rasch model. The item bank provides the basic information to calculate D-scores. The items in the item bank are a subset of all items as collected in builtin_itemtable.

Usage

```
builtin_itembank
```

Format

A `data.frame` with variables:

Name	Label
`key`	String indicating a specific Rasch model
`item`	Item name, gsed lexicon
`tau`	Difficulty estimate
`label`	Label (English)
`instrument`	Instrument code
`domain`	Domain code
`mode`	Administration mode
`number`	Item number

Details

The difficulty estimates were estimated by a Rasch model. The `key` indicates the specific Rasch model used to estimate the difficulty. Strictly speaking, one can only compare D-score calculated from the same `key`.

Note

Updates:

- Dec 01, 2022 - Overwrite labels of gto by correct item order.
- Dec 05, 2022 - Adds key `gsed2212`, adding instruments `gl1` and `gs1`, and defining correct order for `gto`
- Jan 05, 2023 - Adds instrument `gh1` to key `gsed2212`

See Also

`dscore()`, `get_tau()`, `builtin_itemtable()`

Examples

```
# count number of items per instrument in each key
table(builtin_itembank$instrument, builtin_itembank$key)
```

builtin_itemtable *Collection of items from instruments measuring early child development*

Description

The built-in variable `builtin_itemtable` contains the name and label of items for measuring early child development.

Usage

```
builtin_itemtable
```

Format

A `data.frame` with variables:

Name	Label
`item`	Item name, gsed lexicon
`equate`	Equate group
`label`	Label (English)

Details

The `builtin_itemtable` is created by script `data-raw/R/save_builtin_itemtable.R`.

Updates:

- May 30, 2022 - added gto (LF) and gpa (SF) items
- June 1, 2022 - added seven gsd items
- Nov 24, 2022 - Added instruments gs1, gs2
- Dec 01, 2022 - Labels of gto replaced by correct order. Incorrect item order affects analyses done on LF between 20220530 - 20221201 !!!
- Dec 05, 2022 - Redefines gs1 and instrument for Phase 2, removes gs2 (139) Adds gl1 (Long Form Phase 2 items 155)
- Jan 05, 2023 - Adds 55 items from GSED-HF

Author(s)

Compiled by Stef van Buuren using different sources

`builtin_keys` *Available keys for calculating the D-score*

Description

A key contains the item difficulty estimates from a given Rasch model. The difficulty estimates (`tau`) are used to calculate D-scores. D-scores can only be compared when calculated with the same key.

Usage

```
builtin_keys
```

Format

`builtin_keys` is a `data.frame` with variables:

Name	Label
`key`	String. Name of the key indicating the Rasch model
`base_population`	String. Name of the base population for the key
`n_items`	Number of items in the key
`n_instruments`	Number of instruments in the key
`intercept`	Intercept to convert logit into D-score
`slope`	Slope to convert logit into D-score
`from`	Starting value of the quadrature points
`to`	Stopping value of the quadrature points
`by`	Increment of the quadrature points
`retired`	Has the key been retired?

Note

20240609 SvB: Added `builtin_keys` table by `data-raw\data\R\save_builtin_keys.R`

`builtin_references` *Collection of age-conditional reference distributions*

Description

A data frame containing the age-dependent distribution of the D-score for children aged 0-5 years. The distribution is modelled after the LMS distribution (Cole & Green, 1992) or BCT model (Stasinopoulos & Rigby, 2022) and is equal for both boys and girls. The LMS/BCT values can be used to graph reference charts and to calculate age-conditional Z-scores, also known as the *Development-for-Age Z-score (DAZ)*.

Usage

```
builtin_references
```

Format

A `data.frame` with the following variables:

Name	Label
`population`	Name of the reference population
`key`	D-score key, e.g., `"dutch"`, `"gcdg"` or `"gsed"`
`distribution`	Distribution family: `"LMS"` or `"BCT"`
`age`	Decimal age in years
`mu`	M-curve, median D-score, P50
`sigma`	S-curve, spread expressed as coefficient of variation
`nu`	L-curve, the lambda coefficient of the LMS/BCT model for skewness
`tau`	Kurtosis parameter in the BCT model
`P3`	P3 percentile
`P10`	P10 percentile
`P25`	P25 percentile
`P50`	P50 percentile
`P75`	P75 percentile
`P90`	P90 percentile
`P97`	P97 percentile
`SDM2`	-2SD centile
`SDM1`	-1SD centile
`SD0`	0SD centile, median
`SDP1`	+1SD centile
`SDP2`	+2SD centile

Details

Here are more details on the reference population: The `"dutch"` references were calculated from the SMOCC data, and cover age range 0-2.5 years (van Buuren, 2014). The `"gcdg"` references were calculated from the 15 cohorts of the GCDG-study, and cover age range 0-5 years (Weber, 2019). The `"phase1"` references were calculated from the GSED Phase 1 validation data (GSED-BGD, GSED-PAK, GSED-TZA) cover age range 2w-3.5 years. The age range 3.5-5 yrs is linearly extrapolated and are only indicative. The `"preliminary_standards"` were calculated from the GSED Phase 1 validation data (GSED-BGD, GSED-PAK, GSED-TZA) using a subset of children with covariate indicating healthy development.

References

Cole TJ, Green PJ (1992). Smoothing reference centile curves: The LMS method and penalized likelihood. Statistics in Medicine, 11(10), 1305-1319.

Van Buuren S (2014). Growth charts of human development. Stat Methods Med Res, 23(4), 346-368. https://stefvanbuuren.name/publication/van-buuren-2014-gc/

Weber AM, Rubio-Codina M, Walker SP, van Buuren S, Eekhout I, Grantham-McGregor S, Caridad Araujo M, Chang SM, Fernald LCH, Hamadani JD, Hanlon A, Karam SM, Lozoff B, Ratsifandrihamanana L, Richter L, Black MM (2019). The D-score: a metric for interpreting the early development of infants and toddlers across global settings. BMJ Global Health, BMJ Global Health 4: e001724. `https://gh.bmj.com/content/bmjgh/4/6/e001724.full.pdf`

Stasinopoulos M, Rigby R (2022). gamlss.dist: Distributions for Generalized Additive Models for Location Scale and Shape, R package version 6.0-3, `https://CRAN.R-project.org/package=gamlss.dist`

See Also

`dscore()`

Examples

```
# get an overview of available references per key
table(builtin_references$population, builtin_references$key)
```

`calculate_posterior` *Calculate posterior of ability*

Description

If the tauj is not within the range rello - relhi from the dynamic EAP, the procedure ignores the score of item j.

Usage

```
calculate_posterior(scores, tau, qp, scale, mu, sd, relhi, rello)
```

Arguments

`scores` A vector with PASS/FAIL observations. Scores are coded numerically as `pass = 1` and `fail = 0`.

`tau` A vector containing the item difficulties for the item scores in `scores` estimated from the Rasch model in the preferred metric/scale.

`qp` Numeric vector of equally spaced quadrature points.

`scale` Scale expansion

`mu` Numeric scalar. The mean of the prior.

`sd` Numeric scalar. Standard deviation of the prior.

`relhi` Positive numeric scalar. Upper end of the relevance interval

`rello` Negative numeric scalar. Lower end of the relevance interval

Value

A `list` with three elements:

Name	Label
`eap`	Mean of the posterior
`gp`	Vector of quadrature points
`posterior`	Vector with posterior distribution.

Since `dscore V40.1` the function does not return the `"start"` element.

Author(s)

Stef van Buuren, Arjan Huizing, 2020

`count_mu` *Median D-score from the default references for the given key*

Description

Returns the age-interpolated median of the D-score of the default reference for a given key.

Usage

```
count_mu(t, key, prior_mean_NA = NA_real_)
```

Arguments

`t`	Decimal age, numeric vector
`key`	Character, key of the reference population
`prior_mean_NA`	Numeric, prior mean when age is missing

Details

Do not use this function if you want the median D-score for a specific reference.

DEPRECATED in dscore 1.9.6

Value

A vector of length `length(t)` with the median of the default reference population for the key.

count_mu_dutch *Median of Dutch references*

Description

Returns the age-interpolated median of the Dutch references (van Buuren 2014). The working range is 0-3 years. This function is used to set prior mean under key "dutch".

Usage

```
count_mu_dutch(t)
```

Arguments

t	Decimal age, numeric vector

Value

A vector of length `length(t)` with the median of the Dutch references.

Note

Internal function. Called by `dscore()`

Examples

```
dscore:::count_mu_dutch(0:2)
```

count_mu_gcdg *Median of GCDG references*

Description

Returns the age-interpolated median of the GCDG references (Weber et al, 2019). The working range is 0-4 years. This function is used to set prior mean under keys "gcdg" and "gsed1912".

Usage

```
count_mu_gcdg(t)
```

Arguments

t	Decimal age, numeric vector

Value

A vector of length `length(t)` with the median of the GCDG references.

Note

Internal function. Called by `dscore()`

Examples

```
dscore:::count_mu_gcdg(0:2)
```

`count_mu_phase1` *Median of phase1 references*

Description

Returns the age-interpolated median of the phase1 references based on LF & SF in GSED-BGD, GSED-PAK, GSED-TZA. This function is used to set prior mean under keys `"293_0"` and `"gsed2212"`.

Usage

```
count_mu_phase1(t)
```

Arguments

`t`	Decimal age, numeric vector

Details

The interpolation is done in two rounds. First round: Calculate D-scores using .gcdg prior-mean, calculate reference, estimate round 1 parameters used in this function. Round 2: Calculate D-score using round 1 estimates as the prior mean (most differences are within 0.1 D-score points), recalculate references, estimate round 2 parameters used in this function.

Round 1: Count model: <= 9MN: 21.3449 + 26.4916 t + 7.0251(t + 0.2) Count model: > 9Mn & <= 3.5 YR: 14.69947 - 12.18636 t + 69.11675(t + 0.92) Linear model: > 3.5 YRS: 61.40956 + 3.80904 t

Round 2: Count model: < 9MND: 20.5883 + 27.3376 t + 6.4254(t + 0.2) Count model: > 9MND & < 3.5 YR: 14.63748 - 12.11774 t + 69.05463(t + 0.92) Linear model: > 3.5 YRS: 61.37967 + 3.83513 t

The working range is 0-3.5 years. After the age of 3.5 years, the function will increase at an arbitrary rate of 3.8 D-score points per year.

Value

A vector of length `length(t)` with the median of the GCDG references.

Note

Internal function. Called by `dscore()`

Author(s)

Stef van Buuren, on behalf of GSED project

Examples

```
dscore:::count_mu_phase1(0:5)
```

---

count_mu_preliminary_standards *Median of preliminary_standards*

Description

Returns the age-interpolated median of the preliminary_standards based on LF & SF in GSED-BGD, GSED-PAK, GSED-TZA. This function is used to set prior mean under key `"gsed2406"`.

Usage

```
count_mu_preliminary_standards(t)
```

Arguments

`t`	Decimal age, numeric vector

Value

A vector of length `length(t)` with the median of the GCDG references.

Note

Internal function. Called by `dscore()`

Author(s)

Stef van Buuren, on behalf of GSED project

Examples

```
dscore:::count_mu_preliminary_standards(0:5)
```

daz *Calculate Development-for-Age Z-score (DAZ)*

Description

The daz() function calculated the Development-for-Age Z-score (DAZ). The DAZ represents a child's D-score after adjusting for age by an external age-conditional reference.

Usage

```
daz(d, x, reference_table = NULL, dec = 3, verbose = FALSE)

zad(z, x, reference_table = NULL, dec = 2, verbose = FALSE)
```

Arguments

d	Vector of D-scores
x	Vector of ages (decimal age)
reference_table	A data.frame with the LMS or BCT reference values. The default NULL selects the default reference belonging to the key, as specified in the base_population field in dscore::builtin_keys.
dec	The number of decimals (default dec = 3).
verbose	Print out the used reference table (default verbose = FALSE).
z	Vector of standard deviation scores (DAZ)

Details

The zad() is the inverse of daz(): Given age and the Z-score, it finds the raw D-score.

Note 1: The Box-Cox Cole and Green (BCCG) and Box-Cox t (BCT) distributions model only positive D-score values. To increase robustness, the daz() and zad() functions will round up any D-scores lower than 1.0 to 1.0.

Note 2: The daz() and zad() function call modified version of the pBCT() and qBCT() functions from gamlss for better handling of NA's and rounding.

Value

Unnamed numeric vector with Z-scores of length length(d).

Unnamed numeric vector with D-scores of length length(z).

Author(s)

Stef van Buuren

References

Cole TJ, Green PJ (1992). Smoothing reference centile curves: The LMS method and penalized likelihood. Statistics in Medicine, 11(10), 1305-1319.

See Also

dscore()

Examples

```
# using default reference and key
daz(d = c(35, 50), x = c(0.5, 1.0))

# print out names of the used reference table
daz(d = c(35, 50), x = c(0.5, 1.0), verbose = TRUE)

# using the default reference in key gcdg
reftab <- get_reference(key = "gcdg")
daz(d = c(35, 50), x = c(0.5, 1.0), reference_table = reftab)

# using Dutch reference in default key
reftab <- get_reference(population = "dutch", verbose = TRUE)
daz(d = c(35, 50), x = c(0.5, 1.0), reference_table = reftab)
# population median at ages 0.5, 1 and 2 years, default reference
zad(z = rep(0, 3), x = c(0.5, 1, 2))

# population median at ages 0.5, 1 and 2 years, gcdg key
reftab <- get_reference(key = "gcdg", verbose = TRUE)
zad(z = rep(0, 3), x = c(0.5, 1, 2), reference_table = reftab)

# population median at ages 0.5, 1 and 2 years, dutch key
reftab <- get_reference(key = "dutch", verbose = TRUE)
zad(z = rep(0, 3), x = c(0.5, 1, 2), reference = reftab)
```

---

decompose_itemnames *Decomposes item names into their four components*

Description

This utility function decomposes item names into components: instrument, domain, mode and number

Usage

```
decompose_itemnames(x)
```

Arguments

x	A character vector containing item names (gsed lexicon)

Details

The gsed-naming convention is as follows. Position 1-3 codes the instrument, position 4-5 codes the domain, position 6 codes direct/caregiver/message, positions 7-9 is a item sequence number.

Value

A `data.frame` with `length(x)` rows and four columns, named: `instrument`, `domain`, `mode`, and `number`.

Author(s)

Stef van Buuren

References

https://docs.google.com/spreadsheets/d/1zLsSW9CzqshL8ubb7K5R9987jF4YGDVAW_NBw1hR2aQ/edit#gid=0

See Also

`sort_itemnames()`

Examples

```
itemnames <- c("aqigmc028", "grihsd219", "", "by1mdd157", "mdsgmd006")
decompose_itemnames(itemnames)
```

---

dscore *D-score estimation*

Description

The `dscore()` function estimates the following quantities: *D-score*, a numeric score that quantifies child development by one number, *Development-for-Age Z-score (DAZ)* that corrects the D-score for age, *standard error of measurement (SEM)* of the D-score.

Usage

```
dscore(
  data,
  items = names(data),
  key = NULL,
  population = NULL,
  xname = "age",
  xunit = c("decimal", "days", "months"),
  prepend = NULL,
  itembank = NULL,
  metric = c("dscore", "logit"),
  prior_mean = NULL,
  prior_mean_NA = NULL,
  prior_sd = NULL,
  prior_sd_NA = NULL,
  transform = NULL,
  qp = NULL,
  dec = c(2L, 3L),
  relevance = c(-Inf, Inf),
  algorithm = c("current", "1.8.7"),
  verbose = FALSE
)

dscore_posterior(
  data,
  items = names(data),
  key = NULL,
  population = NULL,
  xname = "age",
  xunit = c("decimal", "days", "months"),
  prepend = NULL,
  itembank = NULL,
  metric = c("dscore", "logit"),
  prior_mean = NULL,
  prior_mean_NA = NULL,
  prior_sd = NULL,
  prior_sd_NA = NULL,
  transform = NULL,
  qp = NULL,
  dec = c(2L, 3L),
  relevance = c(-Inf, Inf),
  algorithm = c("current", "1.8.7"),
  verbose = FALSE
)
```

Arguments

`data` A `data.frame` or `matrix` with the data. A row collects all observations made on a child on a set of milestones administered at a given age. The function calculates a D-score for each row. Different rows can correspond to different children or ages.

`items` A character vector containing names of items to be included into the D-score calculation. Milestone scores are coded numerically as `1` (pass) and `0` (fail). By default, D-score calculation is done on all items found in the data that have a difficulty parameter under the specified `key`.

`key` String. They key identifies 1) the difficulty estimates pertaining to a particular Rasch model, and 2) the prior mean and standard deviation of the prior distribution for calculating the D-score. The default key `NULL` sets `key = "gsed2406"`. View `builtin_keys` for an overview of the available keys.

population
String. The name of the reference population to calculate DAZ. Use with(builtin_references, table(key, population)) to see which built-in references are available for key - population combinations. If not specified, the function set the default population as builtin_keys$base_population[key == builtin_keys$key].

xname
A string with the name of the age variable in data. The default is "age". Do not round age.

xunit
A string specifying the unit in which age is measured (either "decimal", "days" or "months"). The default "decimal" corresponds to decimal age in years.

prepend
Character vector with column names in data that will be prepended to the returned data frame. This is useful for copying columns from data into the result, e.g., for matching.

itembank
A data.frame with at least three columns named key, item and tau. By default, the function uses dscore::builtin_itembank. If you specify your own itembank, then you should also provide the relevant transform and qp arguments.

metric
A string, either "dscore" (default) or "logit", signalling the metric in which ability is estimated. daz is not calculated for the logit scale.

prior_mean
NULL (default), a string, a numeric scalar, or a numeric vector with nrow(data) elements. The default value NULL will consult the base_population field in builtin_keys, and use the corresponding median of that reference as prior mean for the D-score. The string should refer to a column name in data that contains user-supplied values of the prior mean for each observation. A numeric scalar will be expanded to all observations. A numeric vector will be used as is.

prior_mean_NA
NULL (default) or a scalar numeric, representing the prior mean for observations with missing ages. By default, D-scores with missing ages will we NA. We suggest setting prior_mean_NA = 50 as a reasonable choice for samples between 0-3 years. The argument is ignored if prior_mean is specified per observation, which gives you full control of priors for observations with missing ages.

prior_sd
NULL (default), a string, a numeric scalar, or a numeric vector with nrow(data) elements. The default (NULL) uses a value of 5 for all ages. The string should refer to a column name in data that contains user-supplied values of the prior sd for each observation. A numeric scalar will be expanded to all observations. A numeric vector will be used as is.

prior_sd_NA
NULL (default) or a scalar numeric, representing the prior sd for observations with missing ages. By default, D-scores with missing ages will we NA. We suggest setting prior_sd_NA = 20 as a reasonable choice for samples between 0-3 years. The argument is ignored if prior_sd is specified per observation, which gives you full control of priors for observations with missing ages.

transform
Numeric vector, length 2, containing the intercept and slope of the linear transform from the logit scale into the the D-score scale. The default (NULL) searches builtin_keys for intercept and slope values.

qp
Numeric vector of equally spaced quadrature points. This vector should span the range of all D-score or logit values. The default (NULL) creates seq(from, to, by) searching the arguments from builtin_keys.

dec
A vector of two integers specifying the number of decimals for rounding the D-score and DAZ, respectively. The default is dec = c(2L, 3L).

`relevance` A numeric vector of length with the lower and upper bounds of the relevance interval. The procedure calculates a dynamic EAP for each item. If the difficulty level (tau) of the next item is outside the relevance interval around EAP, the procedure ignore the score on the item. The default is `c(-Inf, +Inf)` does not ignore scores.

`algorithm` Computational method, for backward compatibility. Either `"current"` (default) or `"1.8.7"` (deprecated).

`verbose` Logical. Print settings.

Details

The scoring algorithm is based on the method by Bock and Mislevy (1982). The method uses Bayes rule to update a prior ability into a posterior ability.

The item names should correspond to the `"gsed"` lexicon.

A key is defined by the set of estimated item difficulties.

Key	Model	Quadrature	Instruments	Direct/Caregiver	Reference
`"dutch"`	`75_0`	`-10:80`	1	direct	Van Buuren, 2014/2020
`"gcdg"`	`565_18`	`-10:100`	13	direct	Weber, 2019
`"gsed1912"`	`807_17`	`-10:100`	21	mixed	GSED Team, 2019
`"293_0"`	`293_0`	`-10:100`	2	mixed	GSED Team, 2022
`"gsed2212"`	`818_6`	`-10:100`	27	mixed	GSED Team, 2022
`"gsed2406"`	`818_6`	`-10:100`	27	mixed	GSED Team, 2024

As a general rule, one should only compare D-scores that are calculated using the same key and the same set of quadrature points. For calculating D-scores on new data, the advice is to use the default, which currently is `"gsed2406"`.

The default starting prior is a mean calculated from a so-called "Count model" that describes mean D-score as a function of age. The The Count models are implemented in the function `[get_mu()]`. By default, the spread of the starting prior is 5 D-score points around the mean D-score, which corresponds to approximately 1.5 to 2 times the normal spread of child of a given age. The starting prior is informative for very short test (say <5 items), but has little impact on the posterior for larger tests.

Value

The `dscore()` function returns a `data.frame` with `nrow(data)` rows. Optionally, the first block of columns can be copied to the result by using `prepend`. The second block consists of the following columns:

Name	Label
`a`	Decimal age (years)
`n`	Number of items with valid (0/1) data
`p`	Percentage of passed milestones
`d`	D-score, mean of posterior distribution
`sem`	Standard error of measurement, standard deviation of the posterior

daz	D-score corrected for age, calculated in Z-scale (for metric "dscore")

The D-score in column d is a linear scale, with values usually ranging from 0 to 100. The D-score is NA if age is missing or if age is lower than -1/12. It is possible to calculate D-scores for cases with missing ages by setting prior_mean_NA and prior_sd_NA to some reasonable value, e.g., prior_mean_NA = 50 and prior_sd_NA = 20, for the sample at hand.

The SEM is a positive number that quantifies the uncertainty of the D-score. It is NA if the D-score is NA.

The DAZ in column daz is a Z-score that corrects the D-score for age. It is NA when there are no reference values for the given age, or when the D-score is extremely unlikely to be valid at the given age.

Advanced applications: The dscore_posterior() function returns a data frame with nrow(data) rows and length(qp) plus prepended columns with the full posterior density of the D-score at each quadrature point. If no valid responses are found, dscore_posterior() returns the prior density. Versions prior to 1.8.5 returned a matrix (instead of a data.frame). Code that depends on the result being a matrix may break and may need adaptation.

Author(s)

Stef van Buuren, Iris Eekhout, Arjan Huizing (2022)

References

Bock DD, Mislevy RJ (1982). Adaptive EAP Estimation of Ability in a Microcomputer Environment. Applied Psychological Measurement, 6(4), 431-444.

Van Buuren S (2014). Growth charts of human development. Stat Methods Med Res, 23(4), 346-368. https://stefvanbuuren.name/publication/van-buuren-2014-gc/

Weber AM, Rubio-Codina M, Walker SP, van Buuren S, Eekhout I, Grantham-McGregor S, Caridad Araujo M, Chang SM, Fernald LCH, Hamadani JD, Hanlon A, Karam SM, Lozoff B, Ratsifandrihamanana L, Richter L, Black MM (2019). The D-score: a metric for interpreting the early development of infants and toddlers across global settings. BMJ Global Health, BMJ Global Health 4: e001724. https://gh.bmj.com/content/bmjgh/4/6/e001724.full.pdf

See Also

builtin_keys(), builtin_itembank(), builtin_itemtable(), builtin_references(), get_tau(), posterior(), milestones()

Examples

```
# using all defaults and properly formatted data
ds <- dscore(milestones)
head(ds)

# step-by-step example
data <- data.frame(
  id = c(
    "Jane", "Martin", "ID-3", "No. 4", "Five", "6",
    NA_character_, as.character(8:10)
  ),
  age = rep(round(21 / 365.25, 4), 10),
  ddifmd001 = c(NA, NA, 0, 0, 0, 1, 0, 1, 1, 1),
  ddicmm029 = c(NA, NA, NA, 0, 1, 0, 1, 0, 1, 1),
  ddigmd053 = c(NA, 0, 0, 1, 0, 0, 1, 1, 0, 1)
)
items <- names(data)[3:5]

# third item is not part of the default key
get_tau(items, verbose = TRUE)

# calculate D-score
dscore(data)

# prepend id variable to output
dscore(data, prepend = "id")

# or prepend all data
# dscore(data, prepend = colnames(data))

# calculate full posterior
p <- dscore_posterior(data)

# check that rows sum to 1
rowSums(p)

# plot full posterior for measurement 7
barplot(as.matrix(p[7, 12:36]),
  names = 1:25,
  xlab = "D-score", ylab = "Density", col = "grey",
  main = "Full D-score posterior for measurement in row 7",
  sub = "D-score (EAP) = 11.58, SEM = 3.99")

# plot P10, P50 and P90 of D-score references
g <- expand.grid(age = seq(0.1, 4, 0.1), p = c(0.1, 0.5, 0.9))
d <- zad(z = qnorm(g$p), x = g$age, verbose = TRUE)
matplot(
  x = matrix(g$age, ncol = 3), y = matrix(d, ncol = 3), type = "l",
  lty = 1, col = "blue", xlab = "Age (years)", ylab = "D-score",
  main = "D-score preliminary standards: P10, P50 and P90")
abline(h = seq(10, 80, 10), v = seq(0, 4, 0.5), col = "gray", lty = 2)

# add measurements made on very preterms, ga < 32 weeks
ds <- dscore(milestones)
points(x = ds$a, y = ds$d, pch = 19, col = "red")
```

get_age_equivalent *Get age equivalents of items that have a difficulty estimate*

Description

This function calculates the ages at which a certain percent in the reference population passes the items.

Usage

```
get_age_equivalent(
  items,
  pct = c(10, 50, 90),
  key = NULL,
  population = NULL,
  transform = NULL,
  itembank = dscore::builtin_itembank,
  xunit = c("decimal", "days", "months"),
  verbose = FALSE
)
```

Arguments

`items`	A character vector containing names of items to be included into the D-score calculation. Milestone scores are coded numerically as `1` (pass) and `0` (fail). By default, D-score calculation is done on all items found in the data that have a difficulty parameter under the specified `key`.
`pct`	Numeric vector with requested percentiles (0-100). The default is `pct = c(10, 50, 90)`.
`key`	String. They key identifies 1) the difficulty estimates pertaining to a particular Rasch model, and 2) the prior mean and standard deviation of the prior distribution for calculating the D-score. The default key `NULL` sets `key = "gsed2406"`. View `builtin_keys` for an overview of the available keys.
`population`	String. The name of the reference population to calculate DAZ. Use `with(builtin_references, table(key, population))` to see which built-in references are available for `key` - `population` combinations. If not specified, the function set the default population as `builtin_keys$base_population[key == builtin_keys$key]`.
`transform`	Numeric vector, length 2, containing the intercept and slope of the linear transform from the logit scale into the the D-score scale. The default (`NULL`) searches `builtin_keys` for intercept and slope values.
`itembank`	A `data.frame` with at least three columns named `key`, `item` and `tau`. By default, the function uses `dscore::builtin_itembank`. If you specify your own `itembank`, then you should also provide the relevant `transform` and `qp` arguments.
`xunit`	A string specifying the unit in which age is measured (either `"decimal"`, `"days"` or `"months"`). The default `"decimal"` corresponds to decimal age in years.
`verbose`	Logical. Print settings.

Value

data.frame with four columns: item, d (D-score), pct (percentile), and a (age-equivalent, in xunit units).

Note

The function internally defines a scale factor given the key.

Examples

```
get_age_equivalent(c("gpagmc018", "gtogmd026", "ddicmm050"))
```

---

get_itemnames *Extract item names*

Description

The get_itemnames() function matches names against the 9-code template. This is useful for quickly selecting names of items from a larger set of names.

Usage

```
get_itemnames(
  x,
  instrument = NULL,
  domain = NULL,
  mode = NULL,
  number = NULL,
  strict = FALSE,
  itemtable = NULL,
  order = "idnm"
)
```

Arguments

x	A character vector, data.frame or an object of class lean. If not specified, the function will return all item names in itemtable.
instrument	A character vector with 3-position codes of instruments that should match. The default instrument = NULL allows for all instruments.
domain	A character vector with 2-position codes of domains that should match. The default instrument = NULL allows for all domains.
mode	A character vector with 1-position codes of the mode of administration. The default mode = NULL allows for all modes.
number	A numeric or character vector with item numbers. The default number = NULL allows for all numbers.

`strict` A logical specifying whether the resulting item names must conform to one of the built-in names. The default is `strict = FALSE`.

`itemtable` A `data.frame` set up according to the same structure as `builtin_itemtable()`. If not specified, the `builtin_itemtable` is used.

`order` A four-letter string specifying the sorting order. The four letters are: `i` for instrument, `d` for domain, `m` for mode and `n` for number. The default is `"idnm"`.

Details

The gsed-naming convention is as follows. Position 1-3 codes the instrument, position 4-5 codes the domain, position 6 codes direct/caregiver/message, positions 7-9 is a item sequence number.

Value

A vector with names of items

Author(s)

Stef van Buuren 2020

See Also

`sort_itemnames()`

Examples

```
itemnames <- c("aqigmc028", "grihsd219", "", "age", "mdsgmd999")

# filter out impossible names
get_itemnames(itemnames)
get_itemnames(itemnames, strict = TRUE)

# only items from specific instruments
get_itemnames(itemnames, instrument = c("aqi", "mds"))
get_itemnames(itemnames, instrument = c("aqi", "mds"), strict = TRUE)

# get all items from the se domain of iyo instrument
get_itemnames(domain = "se", instrument = "iyo")

# get all item from the se domain with direct assessment mode
get_itemnames(domain = "se", mode = "d")

# get all item numbers 70 and 73 from gm domain
get_itemnames(number = c(70, 73), domain = "gm")
```

get_itemtable *Get a subset of items from the itemtable*

Description

The builtin_itemtable object in the dscore package contains basic meta-information about items: a name, the equate group, and the item label. The get_itemtable() function returns a subset of items in the itemtable.

Usage

```
get_itemtable(items = NULL, itemtable = NULL, decompose = FALSE)
```

Arguments

items	A logical or character vector of item names to return. The default (NULL) returns all items.
itemtable	A data.frame set up according to the same structure as builtin_itemtable(). If not specified, the builtin_itemtable is used. If itemtable = "", then a dynamic item table is created from any specified item names.
decompose	If TRUE, the function adds four columns: instrument, domain, mode and number.

Value

A data.frame with seven columns.

See Also

get_labels(), get_itemnames()

Examples

```
head(get_itemtable(), 3)
get_itemtable(LETTERS[1:3], "")
```

get_labels *Get labels for items*

Description

The get_labels() function obtains the item labels for a specified set of items.

Usage

```
get_labels(items = NULL, trim = NULL, itemtable = NULL)
```

Arguments

items A character vector of item names to return. The default (NULL) returns the labels of all items.

trim The maximum number of characters in the label. The default trim = NULL does not trim labels.

itemtable A data.frame set up according to the same structure as builtin_itemtable(). If not specified, the builtin_itemtable is used.

Value

A named character vector with length(items) elements with item labels, in the same order as in items.

See Also

builtin_itemtable(), get_itemnames()

Examples

```
# get labels of first two Macarthur items
get_labels(get_itemnames(instrument = "mac", number = 1:2), trim = 40)
```

get_mu *Median D-score from the base population for a given key*

Description

Returns the age-interpolated median of the D-score of the default reference for a given key.

Usage

```
get_mu(t, key, prior_mean_NA = NA_real_)
```

Arguments

t Decimal age, numeric vector

key Character, key of the reference population

prior_mean_NA Numeric, prior mean when age is missing

Details

Use get_reference() for more options.

Value

A vector of length length(t) with the median of the default reference population for the key.

get_reference *Get D-score reference*

Description

The get_reference() function selects the D-score reference distribution.

Usage

```
get_reference(
  population = NULL,
  key = NULL,
  references = dscore::builtin_references,
  verbose = FALSE,
  ...
)
```

Arguments

population String. The name of the reference population to calculate DAZ. Use with(builtin_references, table(key, population)) to see which built-in references are available for key - population combinations. If not specified, the function set the default population as builtin_keys$base_population[key == builtin_keys$key].

key String. They key identifies 1) the difficulty estimates pertaining to a particular Rasch model, and 2) the prior mean and standard deviation of the prior distribution for calculating the D-score. The default key NULL sets key = "gsed2406". View builtin_keys for an overview of the available keys.

references A data.frame with the same structure as builtin_references. The default is to use builtin_references.

verbose Logical. Print settings.

... Used to test whether the call contained the deprecated argument references.

Value

A data.frame with the LMS reference values.

Note

No references for population "gsed" exist. The function will silently rewrite population = "gsed" into to the population = "gsed".

The "dutch" reference was published in Van Buuren (2014) The "gcdg" was calculated from 15 cohorts with direct observations (Weber, 2019). The "phase1" references were calculated from the GSED Phase 1 validation data (GSED-BGD, GSED-PAK, GSED-TZA) cover age range 2w-3.5 years. The age range 3.5-5 yrs is linearly extrapolated and are only indicative. The "preliminary_standards" references were calculated from the GSED Phase 1 validation using a subset of children with healthy development.

References

Van Buuren S (2014). Growth charts of human development. Stat Methods Med Res, 23(4), 346-368.

Weber AM, Rubio-Codina M, Walker SP, van Buuren S, Eekhout I, Grantham-McGregor S, Caridad Araujo M, Chang SM, Fernald LCH, Hamadani JD, Hanlon A, Karam SM, Lozoff B, Ratsifandrihamanana L, Richter L, Black MM (2019). The D-score: a metric for interpreting the early development of infants and toddlers across global settings. BMJ Global Health, BMJ Global Health 4: e001724. https://gh.bmj.com/content/bmjgh/4/6/e001724.full.pdf.

See Also

builtin_references()

Examples

```
# see key-population combinations of builtin_references
table(builtin_references$key, builtin_references$population)

# get the default reference
reftab <- get_reference()
head(reftab, 2)

# get the default reference for the key "gsed2212"
reftab <- get_reference(key = "gsed2212", verbose = TRUE)

# get dutch reference for default key
reftab <- get_reference(population = "dutch", verbose = TRUE)

# loading a non-existing reference yields zero rows
reftab <- get_reference(population = "france", verbose = TRUE)
nrow(reftab)
```

---

get_tau *Obtain difficulty parameters from item bank*

Description

Searches the item bank for matching items, and returns the difficulty estimates. Matching is done by item name. Comparisons are done in lower case.

Usage

```
get_tau(
  items,
  key = NULL,
  itembank = dscore::builtin_itembank,
  verbose = FALSE
)
```

Arguments

`items` A character vector containing names of items to be included into the D-score calculation. Milestone scores are coded numerically as `1` (pass) and `0` (fail). By default, D-score calculation is done on all items found in the data that have a difficulty parameter under the specified `key`.

`key` String. They key identifies 1) the difficulty estimates pertaining to a particular Rasch model, and 2) the prior mean and standard deviation of the prior distribution for calculating the D-score. The default key `NULL` sets `key = "gsed2406"`. View `builtin_keys` for an overview of the available keys.

`itembank` A `data.frame` with at least three columns named `key`, `item` and `tau`. By default, the function uses `dscore::builtin_itembank`. If you specify your own `itembank`, then you should also provide the relevant `transform` and `qp` arguments.

`verbose` Logical. Print settings.

Value

A named vector with the difficulty estimate per item with `length(items)` elements.

Author(s)

Stef van Buuren 2020

See Also

`builtin_itembank()`, `dscore()`

Examples

```
# difficulty levels in the GHAP lexicon
get_tau(items = c("ddifmd001", "DDigmd052", "xyz"))
```

`gsample` *Sample of 10 children from the GSED Phase 1 study*

Description

A demo dataset with developmental scores at the item level for 10 random children from the GSED Phase 1 data.

Usage

```
gsample
```

Format

A `data.frame` with 10 rows and 295 variables:

Name	Label
`id`	Integer, child ID
`agedays`	Integer, age in days
`gpalac001`	Integer, Cry when hungry...: 1 = yes, 0 = no, NA = not administered
`gpalac002`	Integer, Look at/focus...: 1 = yes, 0 = no, NA = not administered
`...`	and so on..

There are 138 gpa items (item gpamoc008 (clench fists) removed) from GSED SF and and 155 `gto` items from GSED LF.

See Also

`dscore()`

Examples

```
head(gsample)
```

`milestones` *Outcomes on developmental milestones for preterm-born children*

Description

A demo dataset with developmental scores at the item level for a set of 27 preterm children.

Usage

```
milestones
```

Format

A `data.frame` with 100 rows and 62 variables:

Name	Label
`id`	Integer, child ID
`agedays`	Integer, age in days
`age`	Numeric, decimal age in years
`sex`	Character, "male", "female"
`gagebrth`	Integer, gestational age in days
`ddifmd001`	Integer, Fixates eyes: 1 = yes, 0 = no
`...`	and so on..

See Also

`dscore()`

Examples

```
head(milestones)
```

`normalize` *Normalize distribution*

Description

Normalizes the distribution so that the total mass equals 1.

Usage

```
normalize(d, qp)
```

Arguments

`d`	A vector with `length(qp)` elements representing the unscaled density at each quadrature point.
`qp`	Vector of equally spaced quadrature points.

Value

A vector of `length(d)` elements with the prior density estimate at each quadature point.

Note

: Internal function

Examples

```
dscore:::normalize(c(5, 10, 5), qp = c(0, 1, 2))

sum(dscore:::normalize(rnorm(5), qp = 1:5))
```

`posterior` *Calculate posterior for one item given score, difficulty and prior*

Description

Calculate posterior for one item given score, difficulty and prior

Usage

```
posterior(score, tau, prior, qp, scale)
```

Arguments

`score`	Integer, either 0 (fail) and 1 (pass)
`tau`	Numeric, difficulty parameter
`prior`	Vector of prior values on quadrature points `qp`
`qp`	vector of equally spaced quadrature points
`scale`	expansion relative to the logit scale

Details

This function assumes that the difficulties have been estimated by a binary Rasch model, e.g. by `rasch.pairwise.itemcluster()` of the `sirt` package.

Value

A vector of length `length(prior)`

Note

: Internal function

Author(s)

Stef van Buuren, Arjan Huizing, 2020

See Also

`dscore()`

rename_gcdg_gsed *Rename items from gcdg into gsed lexicon*

Description

Function `rename_gcdg_gsed()` translates item names in the gcdg lexicon to item names in the gsed lexicon.

Usage

```
rename_gcdg_gsed(x, copy = TRUE)
```

Arguments

`x`	A character vector containing item names in the gcdg lexicon
`copy`	A logical indicating whether any unmatches names should be copied (`copy = TRUE`) or set to an empty string.

Details

The gsed-naming convention is as follows. Position 1-3 codes the instrument, position 4-5 codes the domain, position 6 codes direct/caregiver/message, positions 7-9 is a item sequence number.

The function currently support ASQ-I (aqi), Barrera-Moncade (bar), Batelle (bat), Bayley I (by1), Bayley II (by2), Bayley III (by3), Dutch Development Instrument (ddi), Denver (den), Griffith (gri), MacArthur (mac), WHO milestones (mds), Mullen (mul), pegboard (peg), South African Griffith (sgr), Stanford Binet (sbi), Tepsi (tep), Vineland (vin).

In cases where the domain of the items isn't clear (vin, bar), the domain is coded as 'xx'.

Value

A character vector of length `length(x)` with gcdg item names replaced by gsed item name.

Author(s)

Iris Eekhout, Stef van Buuren

References

https://docs.google.com/spreadsheets/d/1zLsSW9CzqshL8ubb7K5R9987jF4YGDVAW_NBw1hR2aQ/edit#gid=0

Examples

```
from <- c(
  "ag28", "gh2_19", "a14ps4", "b1m157", "mil6",
  "bm19", "a16fm4", "n22", "ag9", "gh6_5"
)
to <- rename_gcdg_gsed(from, copy = FALSE)
to
```

sample_hf *Sample of 10 children from GSED HF*

Description

A demo dataset with developmental scores at the item level for 10 random children from the GSED Phase 1 data.

Usage

```
sample_hf
```

Format

A `data.frame` with 10 rows and 57 variables:

Name	Label
`subjid`	Integer, child ID
`agedays`	Integer, age in days
`hf001`	Integer, ...: 1 = yes, 0 = no, NA = not administered
`hf002`	Integer, ...: 1 = yes, 0 = no, NA = not administered
`...`	and so on..

Sample data for 55 gpa items forming GSED HF V1

See Also

`dscore()`

Examples

```
head(sample_hf)
```

`sample_lf` *Sample of 10 children from gto (LF)*

Description

A demo dataset with developmental scores at the item level for 10 random children from the GSED Phase 1 data.

Usage

```
sample_lf
```

Format

A `data.frame` with 10 rows and 157 variables:

Name	Label
`subjid`	Integer, child ID
`agedays`	Integer, age in days
`lf001`	Integer, ...: 1 = yes, 0 = no, NA = not administered
`lf002`	Integer, ...: 1 = yes, 0 = no, NA = not administered
`...`	and so on..

Sample data for 155 `gto` items from GSED SF

See Also

`dscore()`

Examples

```
head(sample_lf)
```

`sample_sf` *Sample of 10 children from gpa (SF)*

Description

A demo dataset with developmental scores at the item level for 10 random children from the GSED Phase 1 data.

Usage

```
sample_sf
```

Format

A `data.frame` with 10 rows and 141 variables:

Name	Label
`subjid`	Integer, child ID
`agedays`	Integer, age in days
`sf001`	Integer, Cry when hungry...: 1 = yes, 0 = no, NA = not administered
`sf002`	Integer, Look at/focus...: 1 = yes, 0 = no, NA = not administered
`...`	and so on..

Sample data for 139 gpa items from GSED SF

See Also

`dscore()`

Examples

```
head(sample_sf)
```

`sort_itemnames` *Sorts item names according to user-specified priority*

Description

This function sorts the item names according to instrument, domain, mode and number. The user can specify the sorting order.

Usage

```
sort_itemnames(x, order = "idnm")

order_itemnames(x, order = "idnm")
```

Arguments

`x`	A character vector containing item names (gsed lexicon)
`order`	A four-letter string specifying the sorting order. The four letters are: `i` for instrument, `d` for domain, `m` for mode and `n` for number. The default is `"idnm"`.

Value

`sort_itemnames()` return a character vector with `length(x)` sorted elements. `order_itemnames()` return an integer vector of length `length(x)` with positions of the sorted elements.

Author(s)

Stef van Buuren

See Also

`decompose_itemnames()`

Examples

```
itemnames <- c("aqigmc028", "grihsd219", "", "by1mdd157", "mdsgmd006")
sort_itemnames(itemnames)
```

Index