Package ‘emreliability’

December 17, 2025

Title Test Reliability and CSEM in Educational Measurement

Version 1.0.0

Maintainer Huan Liu <liuhuanbnu@gmail.com>

Description Provides functions for computing test reliability and conditional standard error of measurement (CSEM) based on the methods described in the Reliability in Educational Measurement chapter of the 5th edition of ``Educational Measurement'' by Lee and Harris (2025, ISBN:9780197654965).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 3.5)

Imports ggplot2, rlang

LazyData true

NeedsCompilation no

Author Huan Liu [aut, cre, cph],
Won-Chan Lee [aut],
Min Liang [aut]

Repository CRAN

Date/Publication 2025-12-17 14:50:02 UTC

Contents

`alpha` *Cronbach's Coefficient Alpha*

Description

Compute Cronbach's coefficient alpha and the associated standard error of measurement (SEM) for a set of items.

Usage

```
alpha(x)
```

Arguments

`x`	A data frame or matrix containing item responses, with rows as respondents (subjects) and columns as items.

Details

Cronbach's alpha is an estimate of the internal consistency reliability of a test. This implementation:

- removes rows with any missing values using `stats::na.exclude()`,
- computes the sample covariance matrix of the items,
- uses the classical formula
$$\alpha = \frac{k}{k-1}\left(1 - \frac{\sum \sigma_i^2}{\sigma_X^2}\right),$$
where k is the number of items, σ_i^2 are item variances, and σ_X^2 is the variance of the total score,
- computes SEM as $\mathrm{SD}(X)\sqrt{1-\alpha}$.

Value

A named list with the following elements:

alpha Cronbach's coefficient alpha.

sem Standard error of measurement (SEM) based on alpha.

Examples

```
data(data.u)
alpha(data.u)
```

---

csem_binomial *CSEM and CSSEM with Binomial Model*

Description

Compute the conditional standard error of measurement (CSEM) and conditional standard error of scaled scores (CSSEM) under the binomial model.

Usage

```
csem_binomial(ni, ct = NULL)
```

Arguments

`ni`	A single numeric value indicating the number of items.
`ct`	An optional data frame or matrix containing a conversion table with two columns: the first column as raw scores (0 to `ni`) and the second column as scale scores.

Details

Under the binomial model, for a test with n_i items and a true-score proportion π, the distribution of raw scores is assumed to be $\mathrm{Binomial}(n_i, \pi)$. This function treats each possible raw score $k = 0, 1, \ldots, n_i$ as the true-score value (i.e., $\pi_k = k/n_i$) and computes:

- the CSEM of the raw scores; and
- if `ct` is provided, the CSSEM of the scale scores defined in the conversion table.

Value

A list with:

x A vector of raw scores from 0 to `ni`.

csem A vector of CSEM values (on the raw-score metric) for each raw score.

cssem If `ct` is provided, a vector of CSSEM values for the scale scores corresponding to each raw score.

Examples

```
csem_binomial(40)
csem_binomial(40, ct.u)
```

---

csem_compound_binomial *CSEM, CSSEM, and Reliability under the Compound Binomial Model*

Description

Compute the CSEM, CSSEM, and reliability coefficients for raw scores and scaled scores using the full compound binomial error model.

Usage

```
csem_compound_binomial(x, s, ct = NULL, w = NULL)
```

Arguments

x	Examinee-by-item matrix/data frame of item responses, ordered by stratum.
s	Numeric vector of number of items in each stratum. Sum(s) must equal ncol(x).
ct	Optional conversion table with maxZ + 1 rows. The second column is the scale score corresponding to composite score Z = 0, 1, ..., maxZ.
w	Optional numeric vector of weights for each stratum. Defaults to 1 per stratum.

Value

A list containing:

x Raw total scores (row sums of x).

total_scale If `ct` is provided, the composite scale score for each examinee.

csem CSEM on the raw-score metric for each examinee.

cssem If `ct` is provided, CSSEM on the scale-score metric.

reliability_raw Reliability coefficient for raw scores.

reliability_scale If `ct` is provided, reliability coefficient for scale scores.

Examples

```
data(data.m)
data(ct.m)
csem_compound_binomial(data.m, c(13, 12, 6))

csem_compound_binomial(data.m, c(13, 12, 6), ct.m)
```

csem_info *CSEM of IRT Model via Information*

Description

Compute the CSEM for a unidimensional IRT model using either MLE- or EAP-based test information.

Usage

```
csem_info(theta, ip, est = c("MLE", "EAP"))
```

Arguments

theta	A numeric vector (or object coercible to a numeric vector) containing the ability values at which to compute CSEM.
ip	A data frame or matrix of item parameters. Columns are interpreted in the same way as in `info()`: • 3 columns: `b`, `a`, `c` (3PL; `a` on the `D = 1.702` metric), • 2 columns: `b`, `a` (2PL; `c` internally set to 0), • 1 column: `b` (1PL/Rasch; `a = 1`, `c = 0`).
est	A character string specifying the estimation method: `"MLE"` for maximum likelihood or `"EAP"` for empirical Bayes.

Value

A list containing:

- `theta` — vector of ability values.
- `csemMLE` — CSEM values for MLE (if `est = "MLE"`).
- `csemEAP` — CSEM values for EAP (if `est = "EAP"`).

csem_lord *CSEM Lord Method*

Description

Compute Lord's CSEM in classical test theory under the binomial model.

Usage

```
csem_lord(ni)
```

Arguments

`ni`	A numeric value indicating the number of items (must be at least 2).

Value

A list with:

x Vector of raw scores from 0 to `ni`.

csem Vector of Lord CSEM values corresponding to each raw score.

Examples

```
csem_lord(40)
```

`csem_lord_keats` *CSEM: Lord Keats Method*

Description

Compute CSEM using the Lord Keats approach, which rescales Lord's binomial-model CSEM using empirical KR-20 and KR-21 reliability estimates.

Usage

```
csem_lord_keats(x)
```

Arguments

`x`	A data frame or matrix of item responses, with rows as persons and columns as items. Items are assumed to be dichotomous (0/1).

Details

This function first computes Lord's CSEM under the binomial model via `csem_lord(ni)`, where `ni = ncol(x)`. It then rescales the resulting CSEM curve using the ratio

$$\sqrt{\frac{1 - \text{KR-20}}{1 - \text{KR-21}}},$$

where KR-20 and KR-21 are computed from the observed data via `kr20(x)` and `kr21(x)`, respectively.

Value

A list with:

x Vector of raw scores from 0 to `ni`.

csem Vector of CSEM values under the Lord Keats method.

Examples

```
data(data.u)
csem_lord_keats(data.u)
```

`cssem_polynomial` *Polynomial Method for CSSEM*

Description

Implement the polynomial method for computing conditional standard errors of measurement for scale scores (CSSEM). A polynomial regression of scale scores on raw scores is fit for degrees 1 through `K`; for each degree `k`, the transformation derivative is used to map raw-score CSEM values to scale-score CSSEM values.

Usage

```
cssem_polynomial(csemx, ct, K = 10, gra = TRUE)
```

Arguments

`csemx`	A data frame or matrix containing raw scores and their CSEM on the raw-score metric. It must have at least the following numeric columns: • `x`: raw scores, • `csem`: conditional standard errors of measurement on the raw-score metric.
`ct`	A data frame or matrix containing the score conversion table. It must have at least the following numeric columns: • `x`: raw scores (matching those in `csemx`), • `ss`: scale scores corresponding to each raw score.
`K`	Integer. Highest polynomial degree to fit. Defaults to `10`.
`gra`	Logical. If `TRUE`, a plot of the fitted polynomial curve and the observed conversion points is produced for each degree `k`.

Details

At the beginning of the function, `csemx` and `ct` are merged by the `x` column (inner join) to create an internal data frame . Only rows with `x` values present in both inputs are used. The polynomial model is then fit to `ss ~ poly(x, k, raw = TRUE)` for `k = 1, ..., K`.

Value

A list with two components:

rsquared A matrix with one column containing the R-squared values from polynomial fits of degree `k = 1, ..., K`, where `K` is the largest successfully fitted degree.

cssempoly A data frame containing the merged data (`x`, `csem`, `ss`) and, for each degree `k`, the additional columns:

- `fx_k1`, `fx_k2`, ...: transformation derivatives $f'_k(x)$ for each raw score,
- `ss_k1`, `ss_k2`, ...: fitted (rounded) scale scores from the polynomial of degree `k`,
- `cssem_k1`, `cssem_k2`, ...: CSSEM values on the scale-score metric, computed as $f'_k(x)\,\mathrm{CSEM}_x$.

Examples

```
data(ct.u)
cssem_polynomial(as.data.frame(csem_lord(40)), ct.u, K = 4, gra = TRUE)
```

`ct.m` *Conversion table for multidimensional data.*

Description

A dataset containing the conversion table for the multidimensional data, with first column as raw scores and second column as scale scores

Usage

```
ct.m
```

Format

A data frame with 32 rows and 2 variables:

x raw score

ss scale score

`ct.u` *Conversion table for unidimensional data.*

Description

A dataset containing the conversion table for the unidimensional data, with first column as raw scores and second column as scale scores

Usage

```
ct.u
```

Format

A data frame with 41 rows and 2 variables:

x raw score

ss scale score

`data.m` *Multidimensional data*

Description

A dataset containing the responses of 3000 subjects to 31 items on three subscales (13, 12, and 6 items respectively).

Usage

```
data.m
```

Format

A data frame with 3000 rows and 31 numeric variables named `V1–V31`, each representing the response to one item.

`data.u` *Unidimensional data*

Description

A dataset containing the responses of 3000 subjects to 40 items.

Usage

```
data.u
```

Format

A data frame with 3000 rows and 40 numeric variables named `V1–V40`, each representing the response to one item.

feldt *Feldt's Coefficient*

Description

Compute Feldt's coefficient as an estimate of internal consistency reliability.

Usage

```
feldt(x)
```

Arguments

x	A data frame or matrix containing item responses, with rows as subjects and columns as items.

Value

A named list with:

feldt Feldt's coefficient.

Examples

```
data(data.u)
feldt(data.u)
```

info *Information for IRT Model*

Description

Compute test information for a unidimensional IRT model (1PL/2PL/3PL) across a vector of ability values.

Usage

```
info(theta, ip, est = c("MLE", "EAP"), D = 1.702)
```

Arguments

- `theta` Numeric vector of ability values at which to compute test information.
- `ip` A data frame or matrix of item parameters. Columns are interpreted in order as:
  - 3 columns: b, a, c (3PL, with a on the 1.702 metric),
  - 2 columns: b, a (2PL, c internally set to 0),
  - 1 column: b (1PL/Rasch, a = 1, c = 0).
- `est` Character string indicating the estimation method: "MLE" for maximum likelihood or "EAP" for empirical Bayes.
- `D` A numeric constant representing the scaling factor of the IRT model. Defaults to 1.702.

Details

Test information at each θ is the sum of item information. For est = "EAP", this function returns

$$I_{\text{EAP}}(\theta) = I_{\text{MLE}}(\theta) + 1,$$

where the additional 1 reflects the prior (population) contribution under a standard normal prior.

Value

A list with:

theta Vector of ability values.

infoMLE If est = "MLE", vector of test information at each theta.

infoEAP If est = "EAP", vector of test information at each theta.

---

ip.u *Item parameters for unidimensional data.*

Description

A dataset containing the item parameters for the unidimensional data, with first column as b parameters and second column as a parameters

Usage

```
ip.u
```

Format

A data frame with 40 rows and 2 variables:

b b parameter

a a parameter

kr20 *KR-20*

Description

Compute the KR-20 reliability coefficient for dichotomously scored items (e.g., 0/1).

Usage

```
kr20(x)
```

Arguments

`x`	A data frame or matrix of item responses, with rows as persons and columns as items. Items are assumed to be dichotomous (0/1).

Details

KR-20 is an internal consistency reliability estimate for tests with dichotomously scored items. Rows containing missing values are removed using `stats::na.exclude()`.

Value

A single numeric value: the KR-20 reliability coefficient.

Examples

```
data(data.u)
kr20(data.u)
```

kr21 *KR-21*

Description

Compute the KR-21 reliability coefficient for dichotomously scored items (0/1), assuming equal item difficulty.

Usage

```
kr21(x)
```

Arguments

`x`	A data frame or matrix of item responses, with rows as persons and columns as items. Items are assumed to be dichotomous (0/1).

Details

KR-21 is a simplified alternative to KR-20, assuming equal item difficulty. Rows containing missing values are removed using `stats::na.exclude()`.

Value

A single numeric value: the KR-21 reliability coefficient.

Examples

```
data(data.u)
kr21(data.u)
```

`lord_wingersky` *Lord-Wingersky Recursive Formula*

Description

Compute the raw score distribution for a given theta value using the Lord-Wingersky recursive formula, given item-level probabilities of a correct response.

Usage

```
lord_wingersky(probs)
```

Arguments

`probs`	A numeric vector (or matrix) of probabilities that a given theta value will correctly answer each item. If a matrix is provided, it will be coerced to a numeric vector.

Value

A list with:

x Vector of possible raw scores, from 0 to `ni`.

probability Vector of probabilities for each raw score.

`normal_quadra` *Gaussian Quadrature Points and Weights*

Description

Generate Gaussian quadrature points and corresponding normalized weights based on the standard normal density over a symmetric interval.

Usage

```
normal_quadra(n, mm)
```

Arguments

n	Integer. Number of quadrature points (must be >= 2).
mm	Numeric. Positive value giving the maximum absolute value of the quadrature nodes (range will be from -mm to +mm).

Value

A list with:

nodes Quadrature nodes from -mm to +mm.

weights Normalized weights proportional to the standard normal density at each node.

Examples

```
normal_quadra(41, 5)
```

`rel_info` *Marginal Reliability of a Unidimensional IRT Model*

Description

Compute marginal reliability for a unidimensional IRT model using either MLE-based or EAP-based information, via Gaussian quadrature over a standard normal ability distribution.

Usage

```
rel_info(ip, est)
```

Arguments

ip	A data frame or matrix of item parameters with columns in the order `b`, `a`, `c`, where `a` is on the `D = 1.702` metric. If only 1 or 2 columns are supplied, the `info()` function is expected to treat them as 1PL/2PL accordingly.
est	A character string specifying the ability estimation method: `"MLE"` for maximum likelihood or `"EAP"` for empirical Bayes.

Details

Gaussian quadrature with 41 nodes on [-5, 5] is used to approximate the integrals.

Value

A single numeric value: the marginal reliability (MLE or EAP, depending on est).

Examples

```
data(ip.u)
rel_info(ip.u, "MLE")
```

---

rel_test   *Test Reliability and CSEMs for IRT Scores*

Description

Compute test reliability for raw scores (and optionally scale scores), along with associated conditional standard errors of measurement (CSEMs), for a unidimensional IRT model.

Usage

```
rel_test(ip, ct = NULL, nq = 11, D = 1.702)
```

Arguments

ip — A data frame or matrix of item parameters. Columns are interpreted in order as:

- 3 columns: b, a, c (3PL; a on the D metric),
- 2 columns: b, a (2PL; c internally set to 0),
- 1 column: b (1PL/Rasch; a = 1, c = 0).

ct — Optional. A data frame or matrix containing the score conversion table. If supplied, it must have ni + 1 rows (for raw scores 0:ni) and a column named ss giving the corresponding scale scores. If ct = NULL (default), only raw-score reliability and CSEMs are computed.

nq — Integer. Number of quadrature points used to approximate the standard normal ability distribution. Defaults to 11.

D — Numeric. Scaling constant for the logistic IRT model. Defaults to 1.702.

Value

A list with three components:

fx A data frame containing the estimated marginal score distribution for raw scores (and scale scores if `ct` is provided).

rel A data frame with overall error variance, true score variance, observed score variance, and reliability for raw scores, and additionally for scale scores if `ct` is provided.

csem A data frame with theta, weights, expected raw scores and corresponding CSEMs. If `ct` is provided, expected scale scores and scale-score CSEMs are also included.

Examples

```
data(ip.u)
data(ct.u)
rel_test(ip.u)
rel_test(ip.u, ct.u)
```

`spearman_brown` *Spearman–Brown Prophecy Formula*

Description

Compute the predicted test reliability after changing test length, or compute the required test-length ratio to achieve a desired reliability, using the Spearman–Brown prophecy formula.

Usage

```
spearman_brown(rxx, input, type = c("r", "l"))
```

Arguments

`rxx` A numeric value indicating the original reliability (must be between 0 and 1, exclusive).

`input` A numeric value indicating either:

- the ratio of new test length to original test length (if type = "r"), or
- the desired reliability of the new test (if type = "l").

`type` Character string specifying the calculation type:

"r" Compute new reliability given the length ratio.

"l" Compute the length ratio required to achieve a desired reliability.

Details

The Spearman–Brown prophecy formula is:

$$r_{yy} = \frac{kr_{xx}}{1 + (k-1)r_{xx}},$$

where r_{xx} is the original reliability and k is the ratio of the new test length to the original test length. Solving for k gives:

$$k = \frac{r_{yy}(1 - r_{xx})}{r_{xx}(1 - r_{yy})}.$$

Value

A named list depending on `type`:

reliability Predicted reliability of the new test (if type = "r").

ratio Required ratio of new test length to original test length (if type = "l").

Examples

```
spearman_brown(0.7, 3.86, "r")
spearman_brown(0.7, 0.90, "l")
```

`stratified_alpha` *Stratified Cronbach's Coefficient Alpha*

Description

Compute the stratified Cronbach's coefficient alpha for a test composed of several item strata (e.g., subtests or subscales).

Usage

```
stratified_alpha(x, s)
```

Arguments

`x`	A data frame or matrix containing item responses, with rows as subjects and columns as items. Items are assumed to be ordered by stratum.
`s`	A numeric vector giving the number of items in each stratum. The sum of `s` must equal `ncol(x)`.

Details

Stratified alpha is an estimate of the internal consistency reliability of a composite test formed by multiple item strata (e.g., subtests). Each stratum reliability is computed using `alpha()`, and combined using the classical stratified-alpha formula.

Value

A named list with:

stratified_alpha Stratified Cronbach's coefficient alpha.

Examples

```
data(data.m)
stratified_alpha(data.m, c(13, 12, 6))
```

`stratified_feldt` *Stratified Feldt's Coefficient*

Description

Compute the stratified Feldt's coefficient for a test composed of several item strata (e.g., subtests or subscales).

Usage

```
stratified_feldt(x, s)
```

Arguments

`x`	A data frame or matrix containing item responses, with rows as subjects and columns as items. Items are assumed to be ordered by stratum.
`s`	A numeric vector giving the number of items in each stratum. The sum of `s` must equal `ncol(x)`.

Details

Stratified Feldt's coefficient is an estimate of internal consistency reliability for a composite test formed by multiple strata.

Value

A named list with:

stratified.feldt Stratified Feldt's coefficient.

Examples

```
data(data.m)
stratified_feldt(data.m, c(13, 12, 6))
```

Index