Package 'csemTools'

July 29, 2026

Type Package

Title Conditional Standard Error of Measurement Tools for Test Scores

Version 0.1.4

Description Compute and compare conditional
standard errors of measurement (CSEM) across score distributions
using methods from classical test theory. Includes approaches for
smoothing, bootstrapped CSEM, standardized CSEM, CSEM for scale scores,
and assessment of properties of split-half scores. Also supports
comparison with global standard errors derived from reliability
coefficients and graphical visualization of CSEM curves and
relative precision across observed score ranges. Some of these implemented
methods are based on work by
Lord (1955) <doi:10.1002/j.2333-8504.1955.tb00054.x>,
Feldt and Qualls (1996) <doi:10.1111/j.1745-3984.1996.tb00486.x>,
McNeish and Dumas (2025) <doi:10.3758/s13428-025-02611-8>.

URL https://github.com/cmerinos/csemTools

BugReports https://github.com/cmerinos/csemTools/issues

License GPL-3

Encoding UTF-8

Config/roxygen2/version 8.0.0

Imports boot, ggplot2, pbapply, rlang

Suggests EFA.dimensions, psych, psychTools, MBESS, kSamples, scam,
zoo, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Cesar Merino-Soto [aut, cre]

Maintainer Cesar Merino-Soto <sikayax@yahoo.com.ar>

Repository CRAN

Date/Publication 2026-07-29 16:20:27 UTC

Contents

`checkAlpha` *Compute Cronbach's Alpha for Two Test Halves*

Description

Calculates Cronbach's alpha reliability coefficient for each test half separately, with bootstrap confidence intervals.

Usage

```
checkAlpha(half1, half2, B = 500, conf = 0.95, na.rm = TRUE)
```

Arguments

`half1`	A numeric matrix or data frame containing item scores for the first test half.
`half2`	A numeric matrix or data frame containing item scores for the second test half.
`B`	Integer. Number of bootstrap resamples (default = 500).
`conf`	Numeric. Confidence level (default = 0.95).
`na.rm`	Logical. If TRUE (default), rows with any missing values in either half are removed.

Details

Cronbach's alpha is computed as:

$$\alpha = \frac{k\bar{r}}{1 + (k-1)\bar{r}}$$

where k is the number of items and $\bar{r}$ is the average inter-item correlation.

Confidence intervals are estimated via percentile bootstrap.

Value

A data frame with columns: `Half` ("Half 1" or "Half 2"), `Coefficient` ("alpha"), `Estimate` (estimated alpha), `lwr.ci`, `upr.ci` (bootstrap confidence interval).

Examples

```
## Load data
library(EFA.dimensions)
data("data_RSE")

## Recode negative items
data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")] <- 5 - data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")]

## Check split: difficulty criteria
RSE.namesHalf <- checkSplit(data = data_RSE, method = "difficulty")

## check Distribution
checkAlpha(half1 = data_RSE[,RSE.namesHalf$half1],
                half2 = data_RSE[,RSE.namesHalf$half2],
                B = 1000, conf = .95)
```

---

checkAngoff *Compute Angoff-Feldt Reliability Coefficient for two test halves*

Description

Calculates the Angoff-Feldt reliability coefficient for two test halves, which is appropriate when the halves may have unequal lengths. Confidence intervals are estimated via bootstrap.

Usage

```
checkAngoff(half1, half2, B = 500, conf = 0.95, na.rm = TRUE)
```

Arguments

half1	A numeric matrix or data frame containing item scores for the first test half.
half2	A numeric matrix or data frame containing item scores for the second test half.
B	Integer. Number of bootstrap resamples (default = 500).
conf	Numeric. Confidence level (default = 0.95).
na.rm	Logical. If TRUE (default), rows with any missing values in either half are removed.

Details

The Angoff coefficient (Angoff, 1953; Feldt, 2002) is defined as:

$$\rho = \frac{4\text{cov}(X_1, X_2)}{\text{Var}(X_1) + \text{Var}(X_2) + 2\text{cov}(X_1, X_2)}$$

where X_1 and X_2 are the total scores of the two halves.

This coefficient is a generalization of the Spearman-Brown formula that does not require equal length halves. For equal lengths, it reduces to the usual Spearman-Brown reliability.

Value

A data frame with columns: `Coefficient` ("Angoff-Feldt"), `Estimate` (estimated reliability), `lwr.ci`, `upr.ci` (bootstrap percentile confidence interval).

References

Angoff, W. H. (1953). Test reliability and effective test length. *Psychometrika*, 18(1), 1-14. Feldt, L. S. (2002). Reliability estimation when a test is split into two parts of unknown effective length. *Measurement in Education*, 15(3), 295-308.

Examples

```
## Load data
library(EFA.dimensions)
data("data_RSE")

## Recode negative items
data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")] <- 5 - data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")]

## Check split: difficulty criteria
RSE.namesHalf <- checkSplit(data = data_RSE, method = "difficulty")

## check Distribution
checkAngoff(half1 = data_RSE[,RSE.namesHalf$half1],
                half2 = data_RSE[,RSE.namesHalf$half2],
                B = 1000, conf = .95)
```

checkCongeneric *Gilmer-Feldt Congeneric Reliability for Each Test Half*

Description

Computes the Gilmer-Feldt congeneric reliability coefficient for each test half separately, with bootstrap confidence intervals. This is a lightweight alternative to McDonald's Omega, suitable for congeneric measures.

Usage

```
checkCongeneric(half1, half2, B = 500, conf = 0.95, na.rm = TRUE)
```

Arguments

half1	A numeric matrix or data frame with item scores for the first half.
half2	A numeric matrix or data frame with item scores for the second half.
B	Integer. Number of bootstrap resamples (default = 500).
conf	Numeric. Confidence level (default = 0.95).
na.rm	Logical. If TRUE, rows with missing values are removed.

Details

The Gilmer-Feldt coefficient (Gilmer & Feldt, 1983) is a classical reliability estimator for congeneric tests, where items measure the same construct but may have different loadings. It is computed from the covariance matrix of the items using a weighting scheme that maximizes internal consistency.

Bootstrap confidence intervals are obtained via the percentile method.

Value

A data frame with columns: `Half` ("Half 1" or "Half 2"), `Coefficient` ("Gilmer-Feldt"), `Estimate` (estimated reliability), `lwr.ci`, `upr.ci` (bootstrap confidence interval).

References

Gilmer, J. S., & Feldt, L. S. (1983). Reliability estimation for a test with parts of unknown lengths. *Psychometrika*, 48(1), 99-111.

Examples

```
set.seed(123)

## Load data
library(EFA.dimensions)
data("data_RSE")
```

```
## Recode negative items
data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")] <- 5 - data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")]

## Check split: difficulty criteria
RSE.namesHalf <- checkSplit(data = data_RSE, method = "difficulty")

## check Distribution
checkCongeneric(half1 = data_RSE[,RSE.namesHalf$half1],
                half2 = data_RSE[,RSE.namesHalf$half2],
                B = 1000, conf = .95)
```

`checkDistribution` *Compare Distributions of Two Test Halves*

Description

Evaluates distributional similarity between two test halves using:

- Anderson-Darling test (from package **kSamples**) - optional,
- Overlapping Index (OVI) based on kernel density estimation,
- Kendall's W (concordance) derived from Spearman's rho.

Usage

```
checkDistribution(half1, half2, B = 500, conf = 0.95, na.rm = TRUE)
```

Arguments

`half1`	A numeric matrix or data frame for the first test half.
`half2`	A numeric matrix or data frame for the second test half.
`B`	Integer. Number of bootstrap resamples for confidence intervals of OVI and Kendall's W (default = 500).
`conf`	Numeric. Confidence level (default = 0.95).
`na.rm`	Logical. If TRUE, rows with missing values are removed.

Details

The **Overlapping Index** is computed as the area under the minimum of the two kernel density estimates (using Sheather-Jones bandwidth). The index ranges from 0 (no overlap) to 1 (identical distributions).

Kendall's W for two judges is derived from Spearman's rank correlation: $W = (1+\rho_{Spearman})/2$. It ranges from 0 (no agreement) to 1 (perfect agreement).

The **Anderson-Darling test** (from **kSamples**) tests the null hypothesis that the two distributions are identical. If the package is not installed, the test is skipped.

Effect sizes (ES):

- Anderson-Darling: $AD/\sqrt{n}$ (standardised statistic).
- OVI: the Overlapping Index itself.
- Kendall's W: the W coefficient.

Value

A data frame with columns: `Test`, `Statistic` (test value, NA when not applicable), `p.value` (only for AD), `lwr.ci`, `upr.ci` (only for OVI and W), `ES` (effect size).

Examples

```
## Load data
library(EFA.dimensions)
data("data_RSE")

## Recode negative items
data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")] <- 5 - data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")]

## Check split: difficulty criteria
RSE.namesHalf <- checkSplit(data = data_RSE, method = "difficulty")

## check Distribution
checkDistribution(half1 = data_RSE[,RSE.namesHalf$half1],
                  half2 = data_RSE[,RSE.namesHalf$half2],
                  B = 1000, conf = .95)
```

`checkLocation` *Robust Location Comparison for Two Test Halves (Yuen's trimmed t-test)*

Description

Compares the central tendency of two test halves using Yuen's robust paired t-test (trimmed means) and computes the robust effect size (AKP) with confidence interval.

Usage

```
checkLocation(half1, half2, trim = 0.2, conf = 0.95, na.rm = TRUE)
```

Arguments

`half1`	A numeric matrix or data frame for the first test half.
`half2`	A numeric matrix or data frame for the second test half.
`trim`	Proportion of observations to trim from each tail (default = 0.2).
`conf`	Numeric. Confidence level (default = 0.95).
`na.rm`	Logical. If TRUE, rows with missing values are removed.

Details

The test statistic follows a t distribution with trimmed degrees of freedom. The effect size (AKP) is defined as the trimmed mean difference divided by the winsorized standard deviation of the differences (Algina, Keselman, & Penfield, 2005). The confidence interval for the effect size is obtained using the noncentral t distribution via the `MBESS` package. If `MBESS` is not installed, the CI is omitted and a warning is issued.

Value

A data frame with columns: `Test`, `Statistic`, `p.value`, `lwr.ci`, `upr.ci` (for trimmed mean difference), `ES`, `ES.lwr.ci`, `ES.upr.ci`.

References

Yuen, K. K. (1974). The two-sample trimmed t for unequal population variances. Biometrika, 61(1), 165-170.

Algina, J., Keselman, H. J., & Penfield, R. D. (2005). An alternative to Cohen's standardized mean difference effect size: a robust parameter and confidence interval in the two independent groups case. Psychological methods, 10(3), 317-328. doi:10.1037/1082989X.10.3.317

Algina, J., Keselman, H. J., & Penfield, R. D. (2005). Effect Sizes and their Intervals: The Two-Level Repeated Measures Case. Educational and Psychological Measurement, 65(2), 241-258. doi:10.1177/0013164404268675

Examples

```
## Load data
library(EFA.dimensions)
data("data_RSE")

## Recode negative items
data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")] <- 5 - data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")]

## Check split: difficulty criteria
RSE.namesHalf <- checkSplit(data = data_RSE, method = "difficulty")

## check Location
checkLocation(half1 = data_RSE[,RSE.namesHalf$half1],
              half2 = data_RSE[,RSE.namesHalf$half2],
              conf = .95)
```

`checkScale` *Compare Scale (Variability) Between Two Test Halves*

Description

Evaluates differences in variability between two test halves using:

- Log-transformed variability ratio (lnVR) with confidence interval.
- Bonett-Seier test for equality of variances (robust to non-normality).

Usage

```
checkScale(half1, half2, conf = 0.95, na.rm = TRUE)
```

Arguments

half1	A numeric matrix or data frame with scores from the first half.
half2	A numeric matrix or data frame with scores from the second half.
conf	Numeric. Confidence level for lnVR interval (default = 0.95).
na.rm	Logical. If TRUE, rows with missing values are removed.

Value

A data frame with columns: lnVR (log ratio of SDs), SE.lnVR, lwr.ci, upr.ci, cor.halves (Pearson correlation between halves), bs.stat (Bonett-Seier chi-square statistic), bs.p (p-value for equality of variances), n (effective sample size after NA removal).

References

Bonett, D. G., & Seier, E. (2002). Confidence intervals for variance and standard deviation ratios. Computational Statistics & Data Analysis, 40(3), 603-608.

Examples

```
## Load data
library(EFA.dimensions)
data("data_RSE")

## Recode negative items
data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")] <- 5 - data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")]

## Check split: difficulty criteria
RSE.namesHalf <- checkSplit(data = data_RSE, method = "difficulty")

## check Distribution
checkScale(half1 = data_RSE[,RSE.namesHalf$half1],
                half2 = data_RSE[,RSE.namesHalf$half2],
                conf = .95)
```

checkSpearmanBrown *Compute Spearman-Brown Reliability Coefficient*

Description

Calculates the Spearman-Brown reliability coefficient for two test halves and estimates confidence intervals using bootstrap resampling.

Usage

```
checkSpearmanBrown(half1, half2, B = 500, conf = 0.95, na.rm = TRUE)
```

Arguments

half1	A numeric matrix or data frame containing item scores for the first test half.
half2	A numeric matrix or data frame containing item scores for the second test half.
B	Integer. Number of bootstrap resamples (default = 500).
conf	Numeric. Confidence level (default = 0.95).
na.rm	Logical. If TRUE (default), rows with any missing values in either half are removed.

Details

The Spearman-Brown coefficient is computed as:

$$r_{SB} = \frac{2r}{1+r}$$

where r is the Pearson correlation between the total scores of the two halves.

Confidence intervals are estimated via percentile bootstrap.

Value

A data frame with columns: `Coefficient` ("Spearman-Brown"), `Estimate` (estimated reliability), `lwr.ci`, `upr.ci` (bootstrap confidence interval).

Examples

```
## Load data
library(EFA.dimensions)
data("data_RSE")

## Recode negative items
data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")] <- 5 - data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")]

## Check split: difficulty criteria
RSE.namesHalf <- checkSplit(data = data_RSE, method = "difficulty")
```

```
## check Distribution
checkSpearmanBrown(half1 = data_RSE[,RSE.namesHalf$half1],
                   half2 = data_RSE[,RSE.namesHalf$half2],
                   B = 1000, conf = .95)
```

---

checkSplit *Split a Test into Two Halves*

Description

Divides the items of a test into two halves using different methods.

Usage

```
checkSplit(
  data,
  method = c("ritc", "difficulty", "random", "optimized"),
  equal.size = TRUE,
  n_iter = 500,
  top_n = 10,
  seed = 123,
  na.rm = TRUE
)
```

Arguments

data	A data frame or matrix of item responses (rows = persons, columns = items).
method	Character: "ritc", "difficulty", "random", or "optimized".
equal.size	Logical. If TRUE, forces halves to have the same number of items.
n_iter	Integer. Number of random splits when method = "optimized".
top_n	Integer. Number of top splits to report in summary when method = "optimized".
seed	Integer for reproducibility.
na.rm	Logical. If TRUE, removes rows with missing values.

Value

A list with components: half1, half2 (vectors of column names), summary (data frame), and method.

Examples

```
## Load data
library(EFA.dimensions)
data("data_RSE")

## Recode negative items
data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")] <- 5 - data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")]

## Check split
checkSplit(data = data_RSE, method = "ritc")
checkSplit(data = data_RSE, method = "random")
checkSplit(data = data_RSE, method = "optimized")
checkSplit(data = data_RSE, method = "difficulty")

## Choosing split by difficulty criteria
RSE.namesHalf <- checkSplit(data = data_RSE, method = "difficulty")

RSE.namesHalf$half1
RSE.namesHalf$half2
```

---

compareCSEM *Compare Conditional SEM with Global SEM*

Description

Compares a conditional standard error of measurement (CSEM) function against the global (constant) SEM. The global SEM is computed from the observed score standard deviation and a reliability coefficient (e.g., alpha or omega). Optionally, confidence intervals for the global SEM can be obtained from user supplied reliability confidence bounds. The function produces a data frame with ratios and, if requested, plots of the CSEM vs. global SEM and/or the ratio.

Usage

```
compareCSEM(
  data,
  raw.score = "raw.score",
  CSEM = "CSEM",
  lwr.ci.CSEM = NULL,
  upr.ci.CSEM = NULL,
  score = NULL,
  csem = NULL,
  lwr.ci = NULL,
  upr.ci = NULL,
  cutoff = NULL,
  sd.score,
  reliability,
```

```
  reliab.lwrci = NULL,
  reliab.uprci = NULL,
  plot = c("none", "all", "csem", "ratio"),
  digits = 3,
  conf.level = 0.95,
  ...
)
```

Arguments

data	A data frame containing at least the columns specified in raw.score and CSEM. Optionally, lwr.ci.CSEM and upr.ci.CSEM can be provided for confidence bands. Ignored if score and csem are provided.
raw.score	Character. Name of the column with observed scores (default = "raw.score").
CSEM	Character. Name of the column with conditional SEM values (default = "CSEM").
lwr.ci.CSEM	Character. Optional name of the column with lower confidence limits for CSEM.
upr.ci.CSEM	Character. Optional name of the column with upper confidence limits for CSEM.
score	Numeric vector of observed scores. If provided, overrides data.
csem	Numeric vector of conditional SEM values. If provided, overrides data.
lwr.ci	Numeric vector of lower confidence limits for CSEM (optional).
upr.ci	Numeric vector of upper confidence limits for CSEM (optional).
cutoff	Numeric vector of score values where vertical lines are added to the plots. Useful for highlighting cut scores or quantiles. Default = NULL.
sd.score	Numeric. Standard deviation of the observed scores (calculated externally from raw.score).
reliability	Numeric. Reliability coefficient (e.g., alpha, omega, Gilmer-Feldt) used to compute global SEM.
reliab.lwrci	Numeric. Optional lower confidence bound for reliability. If provided, must be used with reliab.uprci.
reliab.uprci	Numeric. Optional upper confidence bound for reliability.
plot	Character. Type of plot to produce: "csem" (CSEM vs. global SEM), "ratio" (ratio CSEM/global SEM), "all" (both plots), or "none" (no plots). Default = "none".
digits	Numeric. Number of decimal places for rounding output numeric columns (default = 3).
conf.level	Numeric. Confidence level for intervals (default = 0.95). Used only for labelling.
...	Additional arguments passed to ggplot2::geom_line() and ggplot2::geom_ribbon().

Details

The method for comparing conditional SEM against global SEM is inspired by the reliability representativeness approach described in McNeish and Dumas (2025), adapted here for standard errors of measurement rather than reliability.

The global SEM is defined as:

$$SEM_{global} = SD_{observed} \times \sqrt{1 - reliability}$$

If reliability confidence bounds (`reliab.lwrci`, `reliab.uprci`) are supplied, they are transformed to SEM bounds using the inverse relationship:

$$SEM\_lower = SD_{observed} \times \sqrt{1 - reliability\_upper}$$

$$SEM\_upper = SD_{observed} \times \sqrt{1 - reliability\_lower}$$

because lower reliability implies larger measurement error. These bounds are used to draw a confidence band around the global SEM line.

The ratio is computed as `CSEM / SEM_global`. Values above 1 indicate that the conditional SEM is larger (worse precision) than the global average; values below 1 indicate better precision.

Value

- `data`: A data frame with columns `raw.score`, `CSEM`, optionally `lwr.ci.CSEM`/`upr.ci.CSEM`, and `ratio` (rounded to `digits`).
- `global_sem`: A data frame with two columns: `value` (label) and `estimate` (numeric). Contains `sem.global` and, if provided, `lwr.ci` and `upr.ci`.
- `plot`: A ggplot2 object (or list of two objects if `plot = "all"`); `NULL` if `plot = "none"`.

References

McNeish, D., Dumas, D. (2025). Reliability representativeness: How well does coefficient alpha summarize reliability across the score distribution? *Behavior Research Methods*, 57, 93. doi:10.3758/s13428025026118

Examples

```
# Example using data frame
df <- data.frame(
  raw.score = 10:40,
  CSEM = 2.3 + 0.005 * (10:40 - 25)^2,
  lwr.ci.CSEM = 2.1 + 0.005 * (10:40 - 25)^2,
  upr.ci.CSEM = 2.5 + 0.005 * (10:40 - 25)^2
)
result <- compareCSEM(data = df,
                      sd.score = 6.0,
                      reliability = 0.85,
                      reliab.lwrci = 0.82,
                      reliab.uprci = 0.88,
                      plot = "all",
                      cutoff = c(15, 30))

# Example using vectors directly
scores <- 10:40
csem_vals <- 2.3 + 0.005 * (scores - 25)^2
compareCSEM(data = NULL, score = scores, csem = csem_vals,
```

```
              sd.score = 6.0, reliability = 0.85,
              plot = "csem")
```

`csemBinom` *Binomial Model for Conditional Standard Error of Measurement (CSEM)*

Description

Computes the Conditional Standard Error of Measurement (CSEM) under Lord's (1955, 1957) binomial error model. This formulation treats the test as a random sample of dichotomously scored items measuring a single ability. The method can be applied to both dichotomous and polytomous items by transforming raw scores into binomial-equivalent scores. Optionally, the function computes confidence intervals for the true number-correct score using either the classic CSEM-based method or the Wilson score method.

Usage

```
csemBinom(
  score.type = c("dich", "poly"),
  nitems,
  min.resp = NULL,
  max.resp = NULL,
  csem.method = c("Lord", "binom", "LordKeats"),
  data = NULL,
  rhoxx = NULL,
  ci = FALSE,
  ci.method = c("csem", "wilson"),
  conf.level = NULL,
  digits = 3,
  rho.report = FALSE
)
```

Arguments

`score.type`	Character: `"dich"` (default) or `"poly"`.
`nitems`	Integer. Number of items in the test. If `data` is provided for dichotomous items, this is automatically set to `ncol(data)`.
`min.resp`	Numeric. Minimum response value per item. **Required when** `score.type = "poly"`; ignored for dichotomous.
`max.resp`	Numeric. Maximum response value per item. **Required when** `score.type = "poly"`; ignored for dichotomous.
`csem.method`	Character: `"Lord"` (default), `"binom"` (alias, same as Lord), or `"LordKeats"` (adjusts Lord's CSEM using KR-20/KR-21 or empirical factor).

data	Optional data frame or matrix of dichotomous (0/1) item responses. Required for csem.method = "LordKeats" if rhoxx is not provided. Only used when score.type = "dich".
rhoxx	Optional numeric value (0 < rhoxx < 1). An estimate of test reliability (interpreted as KR-20). Used for csem.method = "LordKeats" when no data is supplied. Applies the empirical adjustment KR-21 ~= 0.8 x KR-20 (Wilson, Downing & Ebel, 1979).
ci	Logical. If TRUE, compute confidence intervals. Default FALSE.
ci.method	Character: "csem" (classic) or "wilson". Default "csem".
conf.level	Numeric vector of confidence levels (e.g., 0.95). Default NULL == 0.95.
digits	Integer for rounding CSEM and confidence limits. Default 3.
rho.report	Logical. If TRUE and csem.method = "LordKeats", the function returns a list with the CSEM table and the reliability values (KR-20 and KR-21, either empirical or approximated). Default FALSE.

Details

Binomial error model (Lord, 1955, 1957): Each examinee has true proportion-correct ϕ; observed number-correct X is Binomial(n,ϕ) with $n =$ nitems. Lord's estimator of the conditional error variance is

$$\widehat{\sigma}^2_{E|X} = \frac{X(n-X)}{n-1},$$

and the CSEM is its square root. The options "Lord" and "binom" yield numerically identical results.

Polytomous items: Raw scores are linearly transformed to a proportion and then multiplied by n to obtain a binomial-equivalent score:

$$\text{equiv} = n \times \frac{\text{raw} - n \cdot \text{min.resp}}{n \cdot (\text{max.resp} - \text{min.resp})}.$$

Lord-Keats method (csem.method = "LordKeats"): Adjusts Lord's CSEM using the ratio $\sqrt{(1-\text{KR20})/(1-\text{KR21})}$. If a data matrix is provided, KR-20 and KR-21 are computed directly. If only rhoxx (interpreted as KR-20) is given, KR-21 is approximated as 0.8 * rhoxx following Wilson, Downing & Ebel (1979). This approximation works for both dichotomous and polytomous (after transformation) cases.

Confidence intervals:

- ci.method = "csem": equiv.score +/- z * CSEM.
- ci.method = "wilson": Wilson interval for proportion $p = equiv.score/n$.

Value

If rho.report = FALSE (default), a data frame with rows for each possible raw score. Columns: raw.score, prop.score, binom.CSEM, and for score.type = "poly" also equiv.score. If ci = TRUE, additional columns lwr.xx, upr.xx. If rho.report = TRUE and csem.method = "LordKeats", a list with two elements: csem_table (the data frame) and rho (a data frame containing the reliability values used).

References

Lord, F. M. (1955). Estimating test reliability. *Educational and Psychological Measurement*, 15, 325-336. doi:10.1002/j.23338504.1955.tb00054.x

Lord, F. M. (1957). Do tests of the same length have the same standard error of measurement? *Educational and Psychological Measurement*, 17, 510-521. doi:10.1177/001316445701700407

Keats, J. A. (1957). Estimation of error variances of test scores. *Psychometrika*, 22(1), 29-41. doi:10.1177/001316445701700407

Wilson, R. A., Downing, S. M., & Ebel, R. L. (1979). An empirical adjustment of the Kuder-Richardson 21 reliability coefficient to better estimate the Kuder-Richardson 20 coefficient (ED173387). https://eric.ed.gov/?id=ED173387

Examples

```
# Dichotomous, Lord method
csemBinom(score.type = "dich",
nitems = 16,
csem.method = "Lord")


library(psychTools)
data(ability)
data.ability <- ability[complete.cases(ability),]

 # Lord-Keats method, more data
 csemBinom(score.type = "dich",
 nitems = 16,
 ci = TRUE,
 csem.method = "LordKeats",
 data = data.ability,
 rho.report = TRUE)

# Lord-Keats method, no data
 csemBinom(score.type = "dich", nitems = 16,
 ci = FALSE,
 csem.method = "LordKeats", rhoxx = .86)

 # "LordKeats", polytomous items
 csemBinom(score.type = "poly",
 nitems = 10,
 min.resp = 0,
 max.resp = 4,
 ci = FALSE,
 csem.method = "LordKeats",
 rhoxx = .89)
```

csemBoots *Bootstrap Conditional Standard Error of Measurement (CSEM) based on Item Sampling*

Description

Estimates the conditional standard error of measurement for each total score using a non parametric bootstrap of items (tasks). This method simulates parallel test forms by resampling items with replacement and computes the standard deviation of the resulting bootstrap scores for each person. The individual standard errors are then averaged within each observed total score to obtain the conditional SEM. The procedure follows the logic of Colton, Gao, & Kolen (1996) but is adapted to continuous total scores (sum of Likert items) rather than discrete performance levels.

Usage

```
csemBoots(
  data,
  B = 500,
  cores = 1,
  bin.score = NULL,
  smooth = FALSE,
  degree = 2,
  full.range = FALSE,
  ci = FALSE,
  conf.level = 0.95,
  digits = 3,
  score.range = NULL,
  na.rm = TRUE
)
```

Arguments

data	A data frame or matrix containing the item responses. Rows represent persons, columns represent items. Items can be dichotomous or polytomous (e.g., Likert scales).
B	Integer. Number of bootstrap replications (default = 500).
cores	Integer. Number of CPU cores to use for parallel processing. If cores = 1 (default), the bootstrap runs sequentially with a progress bar. If cores > 1, a parallel cluster is created and the bootstrap is distributed across the cores. A progress bar is still shown (using pbapply).
bin.score	Integer or NULL. If an integer (e.g., 5), persons are grouped into that many quantile groups based on their total score, and the average CSEM within each group is reported. If NULL (default), CSEM is reported for each distinct observed total score (with at least 2 persons).

smooth	Logical. If TRUE, a polynomial regression of the squared CSEM on total score is fitted, and the smoothed CSEM (square root of the fitted values) is reported. This stabilizes estimates at score levels with few observations. Default = FALSE.
degree	Integer. Polynomial degree used when smooth = TRUE. Default = 2.
full.range	Logical. If TRUE and smooth = TRUE, the smoothed CSEM is evaluated for every integer score in score.range. If smooth = FALSE and full.range = TRUE, missing values are filled by carrying forward the last observed CSEM (na.locf). Requires score.range to be provided. Default = FALSE.
ci	Logical. If TRUE, confidence intervals for the true score are computed as score +/- z * CSEM, assuming normally distributed errors. The intervals are truncated to score.range (if supplied) or to the observed range. Default = FALSE.
conf.level	Numeric. Confidence level for the intervals (default = 0.95).
digits	Integer. Number of decimal places for rounding the output. Default = 3.
score.range	Numeric vector of length 2 (min, max). Required when full.range = TRUE. Defines the theoretical score range (e.g., c(10,50)). Also used to truncate confidence intervals (if ci = TRUE). If NULL, the observed minimum and maximum are used for truncation.
na.rm	Logical. If TRUE (default), rows (persons) with any missing item responses are removed before analysis.

Details

Methodological flow (adapted from Colton et al., 1996):

1. The observed total score is computed for each person as the sum of item responses.
2. For each bootstrap replication b = 1 ... B:
   - A bootstrap sample of **items** (columns) is drawn with replacement, preserving the original number of items.
   - For every person, the total score on the bootstrap item set is computed.
3. After all replications, each person has B bootstrap total scores. The standard deviation of these B scores is the **individual standard error of measurement** ($\hat{\sigma}(\Delta_p)$).
4. Persons are grouped by their **observed total score** (from the original data). Within each score level, the individual error variances are averaged, and the square root is taken: $CSEM(x) = \sqrt{\frac{1}{n_x} \sum_{p:X_p=x} \hat{\sigma}^2(\Delta_p)}$.
5. Optionally, a polynomial smoothing is applied to the squared CSEM values to reduce sampling variability at score levels with few persons.
6. If bin.score is provided, the persons are divided into quantile groups (based on the original total scores) and the average CSEM inside each group is reported.

Why bootstrap items (not persons)?: The standard error of measurement refers to the variability of a person's score across hypothetically parallel test forms. Resampling persons would simulate sampling from the population, not parallel forms. Resampling **items** creates different versions of the test, which is the correct approach to estimate the conditional error variance. This is the essence of the method used by Colton et al. (1996) for performance level scores, here extended to continuous total scores.

Parallel processing and progress bar: When `cores = 1`, the bootstrap runs sequentially and a progress bar is shown via `pbapply::pblapply`. When `cores > 1`, a cluster is created using `parallel::makeCluster`, the bootstrap is distributed with `pbapply::pblapply` (which automatically displays a progress bar for parallel tasks), and the cluster is stopped afterwards. The number of cores should not exceed the available CPU cores.

Handling of missing data and small groups: Persons with any missing item responses are removed if `na.rm = TRUE`. Score levels with fewer than 2 persons receive a missing CSEM (NA) in the raw output; they are omitted from smoothing and from the `full.range` expansion (except when `full.range = TRUE` and `smooth = FALSE`, where NAs are filled by last observation carried forward).

Value

A list with up to two elements:

CSEM A data frame with columns:

- `score` : total score (integer).
- `n` : number of persons with that score.
- `CSEM` (or `CSEM.smooth`) : conditional standard error of measurement.
- `lwr.ci`, `upr.ci` : confidence limits (if `ci = TRUE`).

binned.CSEM If `bin.score` is provided, a data frame with:

- `group` : quantile group number.
- `range` : score range of the group.
- `n` : number of persons in the group.
- `mean_score` : average total score in the group.
- `CSEM.mean` : average CSEM within the group.
- `lwr.ci`, `upr.ci` : confidence limits (if `ci = TRUE`).

References

Colton, D. A., Gao, X., & Kolen, M. J. (1997). Assessing the reliability of performance level scores using bootstrapping. ACT Research Report Series, 97-3. Iowa City, 1A: ACT.

Examples

```
library(psych)

# Loading data
data("bfi")

# Choosing variables
data.bfi <- bfi[, c("N1", "N2", "N3", "N4", "N5", "gender", "age")]

# Clean for missing values
data.bfi.nmiss <- data.bfi[complete.cases(data.bfi), ]

# CSEM with bootstrapping
csemBoots(data.bfi.nmiss[, 1:5], B = 100, cores = 1)
```

```
# CSEM with bootstrapping, smoothing, full range, CI
csemBoots(data.bfi.nmiss[, 1:5], B = 100, smooth = TRUE, full.range = TRUE,
                    score.range = c(10, 50), ci = TRUE, cores = 2)

# Quantile groups (quintiles)
csemBoots(data.bfi.nmiss[, 1:5], B = 100, smooth = TRUE, full.range = TRUE,
                    score.range = c(10, 50), ci = TRUE, cores = 2, bin.score = 5)
```

`csemFeldtQualls` *Feldt & Qualls (1996) method for Conditional Standard Error of Measurement (CSEM)*

Description

Implements the improved method by Feldt & Qualls (1996) for estimating conditional measurement error variance. The test is partitioned into `n.parts` (or as specified by `part_items`). For each person, an unbiased estimate of the error variance is computed from the part-test scores. These individual estimates are then averaged within each total score group to obtain the CSEM.

This function does not apply smoothing; for the smoothed version (polynomial regression) see `csemMF`.

Usage

```
csemFeldtQualls(
  data,
  n.parts = NULL,
  part_items = NULL,
  min.items.per.part = 2,
  bin.score = NULL,
  full.range = FALSE,
  ci = FALSE,
  conf.level = 0.95,
  digits = 3,
  score.range = NULL,
  na.rm = TRUE
)
```

Arguments

`data`	A data frame or matrix with item responses (subjects in rows, items in columns). Items can be dichotomous or polytomous.
`n.parts`	Integer. Number of parts into which the test will be split (by column order, as balanced as possible). Ignored if `part_items` is provided. Default is `NULL`, which with `part_items = NULL` sets `n.parts = ncol(data)` (each item as a part).

part_items	Optional list. Each element is a character vector of column names or an integer vector of column indices defining the items in that part. If provided, n.parts is ignored.
min.items.per.part	Integer. Minimum number of items per part (default 2). A warning is issued if any part has fewer items.
bin.score	Integer. Number of quantile groups (e.g., 5 for quintiles). If NULL (default), CSEM is reported for each observed score (with n >= 2).
full.range	Logical. If TRUE, report all integer scores from score.range[1] to score.range[2] (requires score.range). Missing values are filled with the last valid observation (carried forward). Default FALSE.
ci	Logical. If TRUE, compute confidence intervals for the true score. Default FALSE.
conf.level	Numeric. Confidence level (default 0.95).
digits	Integer. Rounding for CSEM and confidence limits. Default 3.
score.range	Numeric vector of length 2 (min, max). Required for full.range = TRUE. Also used to truncate confidence intervals (if ci = TRUE). If NULL, the observed range is used for truncation.
na.rm	Logical. If TRUE (default), removes rows with any missing values.

Value

A list with elements:

CSEM	data.frame with columns score, n, and CSEM (raw). If ci = TRUE, also lwr.ci and upr.ci (truncated to possible score range).
binned.CSEM	(if bin.score is integer) data.frame with quantile groups: group, range, n, mean_score, CSEM.mean, and intervals if ci = TRUE.

References

Feldt, L. S., & Qualls, A. L. (1996). Estimation of measurement error variance at specific score levels. *Journal of Educational Measurement*, 33(2), 141-156. doi:10.1111/j.17453984.1996.tb00486.x

Examples

```
library(psych)

# Loading data
data("bfi")

# Choosing variables
data.bfi <- bfi[, c("N1", "N2", "N3", "N4", "N5", "gender", "age")]

# Clean for missing values
data.bfi.nmiss <- data.bfi[complete.cases(data.bfi), ]

# CSEM, Feldt-Qualls method
csemFeldtQualls(data.bfi.nmiss[, 1:5])
```

```
# With Quantile groups (quintiles)
csemFeldtQualls(bfi[, 1:5], n.parts = 2, bin.score = 5)

# With ci = TRUE
csemFeldtQualls(bfi[, 1:5], n.parts = 2, bin.score = 5, ci = TRUE, conf.level = .68)
```

`csemFSG` *ANOVA method for Conditional Standard Error of Measurement (CSEM)*

Description

Estimates conditional standard errors of measurement using the variance-components approach described in Feldt, Steffen, & Gupta (1985). This is the method referred to as "ANOVA" in the JASP Reliability module.

For each unique total score (or quantile group), the CSEM is computed as:

$$CSEM = \sqrt{\frac{J}{J-1}\sum_{j=1}^{J} s_j^2}$$

where J is the number of items and s_j^2 is the sample variance of item j within the group (using divisor $n_g - 1$).

The function offers polynomial smoothing (`smooth = TRUE`) and expansion to a full range of integer scores (`full.range = TRUE`). It also allows aggregation into quantile groups (`bin.score`).

Usage

```
csemFSG(
  data,
  bin.score = NULL,
  smooth = FALSE,
  degree = 2,
  full.range = FALSE,
  ci = FALSE,
  conf.level = 0.95,
  digits = 3,
  score.range = NULL,
  na.rm = TRUE
)
```

Arguments

data	A data frame or matrix with item responses (subjects in rows, items in columns). Items can be dichotomous or polytomous.
bin.score	integer. Number of quantile groups (e.g., 5 for quintiles). If NULL (default), CSEM is reported for each observed score (with n >= 2).
smooth	logical. If TRUE, applies polynomial smoothing to the squared CSEM estimates. Default = FALSE.
degree	integer. Polynomial degree used when smooth = TRUE. Default = 2.
full.range	logical. If TRUE and smooth = TRUE, evaluates the smoothed CSEM for every integer score from score.range[1] to score.range[2]. Requires score.range. Default = FALSE.
ci	logical. If TRUE, compute confidence intervals for the true score. Default = FALSE.
conf.level	numeric. Confidence level for intervals (default 0.95).
digits	integer. Rounding for CSEM and confidence limits. Default = 3.
score.range	numeric vector of length 2 (min, max). Required when full.range = TRUE. Defines the theoretical score range (e.g., c(0,36)). Also used to truncate confidence intervals (if ci=TRUE). If NULL, the observed range is used for truncation.
na.rm	logical. If TRUE (default), removes rows with any missing values.

Value

A list with elements:

CSEM	data.frame with columns score, n, and either CSEM (raw) or CSEM.smooth (smoothed). If ci = TRUE, also lwr.ci and upr.ci (truncated to possible score range).
binned.CSEM	(if bin.score is integer) data.frame with quantile groups: group, range, n, mean_score, CSEM.mean, and intervals if ci = TRUE.

References

Feldt, L. S., Steffen, M., & Gupta, N. C. (1985). A comparison of five methods for estimating the standard error of measurement at specific score levels. *Applied Psychological Measurement*, 9(4), 351-361.

Examples

```
library(psych)

# Loading data
data("bfi")

# Choosing variables
data.bfi <- bfi[, c("N1", "N2", "N3", "N4", "N5", "gender", "age")]

# Clean for missing values
data.bfi.nmiss <- data.bfi[complete.cases(data.bfi), ]
```

```
# CSEM, Feldt, Steffen, & Gupta method, no smooting
csemFSG(bfi[, 1:5])

# with smooting
csemFSG(bfi[, 1:5], smooth = TRUE, degree = 2)

# with binned score (quntiles)
csemFSG(bfi[, 1:5], smooth = TRUE, degree = 2, bin.score = 5)

# add confidence intervals
csemFSG(bfi[, 1:5], smooth = TRUE, degree = 2, bin.score = 5, ci = TRUE, conf.level = .68)
```

---

csemMF *Mollenkopf-Feldt method for Conditional Standard Error of Measurement (CSEM)*

Description

Implements the Mollenkopf-Feldt method, which applies polynomial regression to the individual error variance estimates from Feldt & Qualls (1996). This provides smoothed CSEM values along the score scale.

Usage

```
csemMF(
  data,
  n.parts = NULL,
  part_items = NULL,
  min.items.per.part = 2,
  degree = 2,
  bin.score = NULL,
  full.range = FALSE,
  ci = FALSE,
  conf.level = 0.95,
  digits = 3,
  score.range = NULL,
  na.rm = TRUE
)
```

Arguments

data	A data frame or matrix with item responses (subjects in rows, items in columns).
n.parts	Integer. Number of parts into which the test will be split (by column order, as balanced as possible). Ignored if part_items is provided. Default is NULL, which sets n.parts = ncol(data) (each item as a part).

part_items	Optional list. Each element is a character vector of column names or an integer vector of column indices defining the items in that part. If provided, n.parts is ignored.
min.items.per.part	Integer. Minimum number of items per part (default 2). A warning is issued if any part has fewer items.
degree	Integer. Degree of the polynomial (default 2).
bin.score	Integer. Number of quantile groups (e.g., 5 for quintiles). If NULL (default), no binning is performed. If provided, the function returns a data frame binned.CSEM with average CSEM per quantile group.
full.range	Logical. If TRUE, evaluates the smoothed CSEM for every integer score from score.range[1] to score.range[2] (requires score.range).
ci	Logical. If TRUE, compute confidence intervals for the true score.
conf.level	Numeric. Confidence level (default 0.95).
digits	Integer. Rounding for output.
score.range	Numeric vector length 2. Required if full.range = TRUE. Also used to truncate confidence intervals (if ci = TRUE).
na.rm	Logical. Remove rows with missing values.

Value

A list with elements:

CSEM	data.frame with columns score, n, CSEM.smooth, and if ci = TRUE, lwr.ci and upr.ci.
binned.CSEM	(if bin.score is provided) data.frame with quantile groups: group, range, n, mean_score, CSEM.mean, and intervals if ci = TRUE.

References

Mollenkopf, W. G. (1949). Variation of the standard error of measurement. *Psychometrika*, 14(3), 189-229. Feldt, L. S., & Qualls, A. L. (1996). Estimation of measurement error variance at specific score levels. *Journal of Educational Measurement*, 33(2), 141-156.

Examples

```
#Basic use
## Load data
library(EFA.dimensions)
data("data_RSE")

## Recode negative items
data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")] <- 5 - data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")]

## Split in two halves
RSE.namesHalf <- checkSplit(data = data_RSE, method = "difficulty")
```

```
## Items in the halves?
RSE.namesHalf$half1
RSE.namesHalf$half2

## Run Mollenkoptf-Feldt method
csemMF(data = data_RSE,
ci = FALSE,
degree = 3, n.parts = 2)

## 3 parts
csemMF(data = data_RSE,
ci = FALSE,
degree = 3, n.parts = 3)

## CSEM for binned score too
csemMF(data = data_RSE,
ci = FALSE,
degree = 3,
n.parts = 2, bin.score = 4)

## Full report
csemMF(data = data_RSE,
ci = FALSE, conf.level = .90,
degree = 3,
n.parts = 2, bin.score = 4)
```

`csemStrong` *Strong True Score (Compound Binomial) Model for CSEM*

Description

Computes conditional standard errors of measurement (CSEM) under Lord's strong true score theory (compound binomial model) for dichotomously scored items. The estimator follows Lord (1965) as summarized by Tong and Kolen (2018). Optionally, the function computes confidence intervals for the true number-correct score.

Usage

```
csemStrong(
  data,
  score.type = c("dich", "poly"),
  nitems = NULL,
  min.resp = NULL,
  max.resp = NULL,
  bin.score = NULL,
  ci = FALSE,
  ci.method = c("csem", "wilson"),
```

```
  conf.level = NULL,
  digits = 3,
  full.range = TRUE,
  score.range = NULL,
  summary = FALSE,
  na.rm = TRUE
)
```

Arguments

data	A numeric matrix or data frame with examinees in rows and items in columns. For score.type = "dich" entries must be 0/1; for "poly" they range between min.resp and max.resp.
score.type	Character: "dich" (default) or "poly".
nitems	Optional integer. Number of items. If NULL (default), taken as ncol(data).
min.resp	Numeric. Minimum response value per item. **Required when** score.type = "poly".
max.resp	Numeric. Maximum response value per item. **Required when** score.type = "poly".
bin.score	Integer. Number of quantile groups (e.g., 5 for quintiles). If NULL (default), no binning is performed. If provided, the function returns a data frame binned.CSEM with average CSEM per quantile group.
ci	Logical. If TRUE, compute confidence intervals. Default FALSE.
ci.method	Character: "csem" (classic) or "wilson". Default "csem".
conf.level	Numeric vector of confidence levels (e.g., 0.95). Default NULL == 0.95.
digits	Integer for rounding CSEM and confidence limits. Default 3.
full.range	Logical. If TRUE (default), CSEM values are reported for the full score range (all possible scores). For dichotomous, that is 0:nitems; for polytomous, all observed equivalent scores plus the extremes 0 and nitems. If FALSE, only scores observed in data are reported.
score.range	Optional numeric vector of length 2 (min, max). If provided, confidence intervals are truncated to this range. If NULL, the observed score range (or theoretical range for full.range) is used.
summary	Logical. If TRUE, returns a data frame with key parameters (mu_X, var_Xp, var_Xi, mean_pq, correction_factor). Default FALSE.
na.rm	Logical. If TRUE (default), rows with missing values are removed. If FALSE, stops on missing values.

Value

A list with components:

CSEM	Data frame with columns: raw.score (original total), equiv.score (if polytomous), n (frequency), csem.strong, and confidence limits if requested.
binned.CSEM	(if bin.score is provided) Data frame with quantile groups: group, range, n, mean_score, CSEM.mean, and confidence limits if ci = TRUE.
summary	(if summary = TRUE) Data frame with parameters.

References

Lord, F. M. (1965). A strong true score theory, with applications. *Psychometrika*, 30, 239-270. doi:10.1007/BF02289490

Tong, Y. and Kolen, M.J. (2018). Conditional Standard Errors of Measurement. In Wiley StatsRef: Statistics Reference Online (eds N. Balakrishnan, T. Colton, B. Everitt, W. Piegorsch, F. Ruggeri and J.L. Teugels; pp. 1-8). doi:10.1002/9781118445112.stat06376.pub2

Examples

```
# Dichotomous,  strong true score

library(psychTools)
data(ability)
data.ability <- ability[complete.cases(ability),]

csemStrong(score.type = "dich",
data = data.ability,
nitems = 16,
ci = TRUE,
summary = FALSE)

# Dichotomous,  Compound Binomial Model (strong true score), more summary and binned score
csemStrong(score.type = "dich",
data = data.ability,
nitems = 16,
ci = FALSE,
summary = TRUE,
bin.score = 5)

# Polytomous items,
## Load data
library(EFA.dimensions)
data("data_RSE")

## Recode negative items
data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")] <- 5 - data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")]

csemStrong(score.type = "poly",
data = data_RSE,
min.resp = 1,
max.resp = 4,
ci = FALSE,
summary = TRUE,
full.range = TRUE,
bin.score = 4)
```

csemThorndike *Thorndike's method for Conditional Standard Error of Measurement (CSEM)*

Description

Simplified version without manual merging of small groups. Use `smooth = TRUE` for stable estimates across the whole score range. When `smooth = FALSE` and `full.range = TRUE`, missing values (NA) are filled with the last valid observation (carried forward) for better readability.

Usage

```
csemThorndike(
  half1,
  half2,
  bin.score = NULL,
  smooth = FALSE,
  degree = 2,
  full.range = FALSE,
  ci = FALSE,
  conf.level = 0.95,
  digits = 3,
  score.range = NULL,
  na.rm = TRUE
)
```

Arguments

`half1`	data.frame/matrix with first half items.
`half2`	data.frame/matrix with second half items.
`bin.score`	integer (number of quantile groups) or NULL (individual scores).
`smooth`	logical. If TRUE, applies polynomial smoothing.
`degree`	integer. Polynomial degree (used if smooth=TRUE).
`full.range`	logical. If TRUE, report all integer scores from `score.range[1]` to `score.range[2]` (requires `score.range`).
`ci`	logical. If TRUE, compute confidence intervals for true score.
`conf.level`	numeric. Confidence level (default 0.95).
`digits`	integer. Rounding for output.
`score.range`	numeric vector of length 2 (min, max). Required for full.range=TRUE. Also used to truncate confidence intervals (if ci=TRUE).
`na.rm`	Logical. If TRUE, missing values in items are ignored when computing half-test totals (passed to `rowSums`). Rows are not removed. Default TRUE.

Value

A list with elements:

`CSEM`	data.frame with columns `score`, `n`, and either `CSEM` (raw) or `CSEM.smooth` (smoothed). If ci=TRUE, also `lwr.ci` and `upr.ci` (truncated to possible score range).
`binned.CSEM`	(if bin.score is integer) data.frame with quantile groups.

References

Thorndike, R. L. (1951). Reliability. In E. F. Lindquist (Ed.), Educational measurement (pp. 560-620). American Council on Education.

Lee, W., & Harris, D. J. (2025). Reliability in educational measurement. In L. L. Cook & M. J. Pitoniak (Eds.), Educational measurement (5th ed., pp. 277-381). Oxford University Press. doi:10.1093/oso/9780197654965.003.0005

Examples

```
## Load data
library(EFA.dimensions)
data("data_RSE")

## Recode negative items
data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")] <- 5 - data_RSE[c("Q3", "Q5", "Q8", "Q9", "Q10")]

## Choosing split by difficulty criteria
RSE.namesHalf <- checkSplit(data = data_RSE, method = "difficulty")

RSE.namesHalf$half1
RSE.namesHalf$half2

# Thorndike csem, basic ouput
csemThorndike(half1 = data_RSE[, RSE.namesHalf$half1],
half2 = data_RSE[, RSE.namesHalf$half2],
smooth = FALSE,
ci = FALSE)

# Thorndike csem, smoothing and binned score
csemThorndike(half1 = data_RSE[, RSE.namesHalf$half1],
half2 = data_RSE[, RSE.namesHalf$half2],
smooth = TRUE,
degree = 2,
ci = FALSE,
bin.score = 5)

# Thorndike csem, smoothing, binned score, and confidence interval
csemThorndike(half1 = data_RSE[, RSE.namesHalf$half1],
half2 = data_RSE[, RSE.namesHalf$half2],
smooth = TRUE,
degree = 2,
ci = TRUE,
```

```
conf.level = .90,
bin.score = 5)
```

`kr20` *KR-20 Reliability Coefficient for Dichotomous Items*

Description

Computes the Kuder-Richardson formula 20 (KR-20) reliability coefficient for a test with dichotomously scored (0/1) items. This is a measure of internal consistency.

Usage

```
kr20(data)
```

Arguments

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

Details

The KR-20 formula is:

$$KR20 = \frac{k}{k-1}\left(1 - \frac{\sum p_j q_j}{\sigma_X^2}\right)$$

where k is the number of items, p_j is the proportion correct on item j, $q_j = 1 - p_j$, and σ_X^2 is the variance of the total test scores (using population variance divisor N).

Rows with missing values are removed using `na.omit`.

Value

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

kr21 *KR-21 Reliability Coefficient for Dichotomous Items*

Description

Computes the Kuder-Richardson formula 21 (KR-21) reliability coefficient, a simplified version of KR-20 that assumes equal item difficulties. It is easier to compute but generally underestimates KR-20.

Usage

```
kr21(data)
```

Arguments

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

Details

The KR-21 formula is:

$$KR21 = \frac{k}{k-1}\left(1 - \frac{\bar{X}(k-\bar{X})}{k\sigma_X^2}\right)$$

where k is the number of items, $\bar{X}$ is the mean total score, and σ_X^2 is the variance of total scores (population variance).

Rows with missing values are removed using `na.omit`.

Value

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

plotCSEM *Plot Conditional Standard Error of Measurement (CSEM) Curve*

Description

Creates a ggplot2 plot of the CSEM as a function of test scores, or a plot of the observed score with confidence interval band for the true score.

Usage

```
plotCSEM(
  score,
  csem,
  lwr.ci = NULL,
  upr.ci = NULL,
  plot.type = c("csem", "csemscore"),
  cutoff = NULL,
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  color.line = "black",
  color.points = "darkred",
  color.band = "lightblue",
  color.cutoff = "gray50",
  line.type = "solid",
  point.size = 2,
  save.path = NULL,
  width = 8,
  height = 6
)
```

Arguments

score	Numeric vector of scores (x-axis).
csem	Numeric vector of CSEM values (y-axis for "csem" plot). For "csemscore", this vector is used as the observed score (line diagonal).
lwr.ci	Numeric vector of lower confidence limits (required for "csemscore").
upr.ci	Numeric vector of upper confidence limits (required for "csemscore").
plot.type	Character: "csem" (CSEM curve) or "csemscore" (confidence band).
cutoff	Numeric vector of score values where vertical lines are added. Useful for highlighting cut scores or quantiles. Default = NULL.
title	Character. Plot title. If NULL, a default title is generated.
xlab	Character. X-axis label. If NULL, defaults to "Score".
ylab	Character. Y-axis label. If NULL, auto-generated.
color.line	Color for the line (default = "black").
color.points	Color for points (default = "darkred").
color.band	Color for confidence band/ribbon (default = "lightblue").
color.cutoff	Color for cutoff lines (default = "gray50").
line.type	Line type (default = "solid").
point.size	Size of points (default = 2).
save.path	Optional file path to save plot as PNG. If NULL, plot is not saved.
width	Numeric. Width of saved plot in inches (default = 8).
height	Numeric. Height of saved plot in inches (default = 6).

Value

A ggplot2 object (invisibly). The plot is also printed.

Examples

```
basicData <- csemBinom(score.type = "dich",
nitems = 10)

plotCSEM(score = basicData$raw.score, csem = basicData$binom.CSEM)


# Example with simulated data
scores <- 0:20
csem_vals <- 1 + 0.5 * abs(scores - 10) / 10

# CSEM curve
plotCSEM(score = scores, csem = csem_vals, plot.type = "csem")

# Confidence band for true score (needs lwr.ci and upr.ci)
lwr <- scores - 1.96 * csem_vals
upr <- scores + 1.96 * csem_vals
plotCSEM(score = scores, csem = scores, lwr.ci = lwr, upr.ci = upr,
         plot.type = "csemscore", cutoff = c(5, 10, 15))
```

---

polynomialDegree *Selecting the Polynomial Degree for CSEM Smoothing (Linear Regression)*

Description

Fits polynomial models of degree 1 to `max_degree` to the data `(x, y)` and returns a table containing information criteria (AIC, BIC), adjusted R^2, and the significance of the term of the highest degree. This helps in choosing the appropriate degree to use in `csemFeldt(..., smooth = TRUE, degree = ...)`.

Usage

```
polynomialDegree(
  x,
  y,
  min_degree = 1,
  max_degree = 6,
  show.coefs = FALSE,
  plot = FALSE
)
```

Arguments

x	Numeric vector of scores (cluster centers).
y	Numeric vector of CSEM.raw (unsmoothed estimates).
min_degree	Minimum degree to evaluate (default 1).
max_degree	Maximum degree to evaluate (default 6).
show.coefs	If TRUE, displays the coefficients for each model.
plot	If TRUE, generates a plot of AIC and BIC versus degree.

Value

A list containing:

summary	A data.frame with degrees of freedom, AIC, BIC, adjusted R^2, p-value of the term with the highest degrees of freedom (compared to the previous model), and whether the degrees of freedom are "better" according to AIC or BIC.
models	A list of lm models for each degrees of freedom.

Examples

```
set.seed(123)
x <- 0:20
y <- 2 + 0.5*x - 0.02*x^2 + rnorm(21, sd=0.2)

res <- polynomialDegree(x, y, max_degree = 4, plot = TRUE)

print(res$summary)
```

---

scaleCSEM *Compute Conditional Standard Error of Measurement (CSEM) for Non-Linear Scale Scores*

Description

Applies the formal methodologies proposed by Feldt and Qualls (1998) to translate raw score CSEMs into non-linear scale score metrics. Supports both the calculus-based Polynomial Method (using monotonic splines) and the interval-based Approximation Method.

Usage

```
scaleCSEM(
  raw,
  scale,
  csem,
  method = c("approx", "polym"),
```

```
  C = NULL,
  plot = FALSE,
  plot.what = "both"
)
```

Arguments

raw	Numeric vector. Raw scores (must be integer values, ideally consecutive).
scale	Numeric vector. Scale scores corresponding to each raw score.
csem	Numeric vector. Conditional standard errors of measurement in raw score units.
method	Character. "approx" (interval method) or "polym" (monotonic spline method).
C	Integer. Interval width for "approx". If NULL, uses round(1.5 * mean(csem)).
plot	Logical. If TRUE, generates a plot.
plot.what	Character. "both", "raw", or "scale" to choose what to display.

Details

Important technical notes:

1. **Use of SCAM in the "polym" method:** This method uses the scam package (Shape Constrained Additive Models) with a monotonic increasing P-spline basis (bs = "mpi"). This choice mathematically enforces a strictly non-negative first derivative across the entire raw score range, matching the psychometric requirement stated by Feldt & Qualls (1998, p. 163). Unconstrained splines (smooth.spline) or high-degree polynomials can produce negative slopes at the boundaries, leading to invalid negative scale score standard errors. The scam approach avoids arbitrary post-hoc adjustments. *Note:* The scam package requires mgcv, which will be installed automatically when you install scam.
2. **Completeness of the conversion table (method "approx"):** This method directly looks up the scale scores associated with $X_0 \pm C$ in the provided vectors. Therefore, it is mandatory that the raw vector contains all integer values required to cover $X_0 \pm C$ for every row within the observed range of raw scores. Ideally, the table should include all integer raw scores from the minimum to the maximum observed. If any required raw score is missing, the function stops with an informative error. No interpolation is performed.

Dependencies: The "polym" method requires the scam package (which in turn depends on mgcv). Please ensure both are installed: install.packages(c("scam", "mgcv")). If these packages are not available, use method = "approx" instead.

Range of raw scores: The function does not assume that raw scores start at 0. It uses the minimum and maximum values present in the raw vector as the natural boundaries of the score scale. This makes it suitable for Likert-type scales (e.g., summing 5 items each scored 1-5 gives a minimum of 5). Internally, the intervals $X_0 \pm C$ are truncated to the observed range $[min(raw), max(raw)]$.

Value

A data.frame with columns: raw, scale, csem, slope, scale_csem.

References

Feldt, L. S., & Qualls, A. L. (1998). Approximating Scale Score Standard Error of Measurement From the Raw Score Standard Error. Applied Measurement in Education, 11(2), 159-177.

Examples

```
# Example with linear transformation (slope = 5)
raw <- 0:10
scale <- seq(20, 70, by = 5)
csem <- c(2.0, 1.8, 1.6, 1.4, 1.3, 1.2, 1.3, 1.4, 1.6, 1.8, 2.0)

scaleCSEM(raw, scale, csem, method = "approx", plot = TRUE)



 # Simulate a nonlinear scale (example: square root)
raw <- 0:10
scale <- round(20 + 30 * sqrt(raw/10))  # no lineal
csem <- c(2.0, 1.8, 1.6, 1.4, 1.3, 1.2, 1.3, 1.4, 1.6, 1.8, 2.0)

scaleCSEM(raw, scale, csem, method = "approx", plot = TRUE)



# Full workflow

# Loading data
library(psych)
data("bfi")

# Choosing variables
data.bfi <- bfi[, c("N1", "N2", "N3", "N4", "N5", "gender", "age")]

# Clean for missing values
data.bfi.nmiss <- data.bfi[complete.cases(data.bfi), ]

# CSEM with bootstrapping
output.boots1 <- csemBoots(data = data.bfi.nmiss[,1:5], ci = FALSE,
conf.level = .90,
full.range = TRUE,
score.range = c(5, 30),
smooth = TRUE, B = 2000)

# Score sum
uno <- table(rowSums(data.bfi.nmiss[,1:5]))

# t Scores, linear transformation
dos <- table(round(psych::rescale(x = rowSums(data.bfi.nmiss[,1:5]), mean = 50, sd = 10)))

# merge ot dataframe
dframe <- cbind.data.frame(score = as.data.frame(uno)$Var1,
```

```
tScore = as.data.frame(dos)[1])

colnames(dframe)[2] <- "tscore"

scaleCSEM(raw = output.boots1$CSEM$score,
scale = as.numeric(dframe$tscore),
csem = output.boots1$CSEM$CSEM.smooth,
method = "polym", C = 5,plot = TRUE, plot.what = "both")
```

`scoreCSEM` *Obtain CSEM for individual scores from a reference table*

Description

Given vectors of reference scores and their CSEM values, this function matches each individual score exactly to a reference score. If an individual score is not found in the reference, a warning is issued and NA is returned. Optionally, it computes confidence intervals assuming normality.

Usage

```
scoreCSEM(score.indiv, score.ref, csem, ci = FALSE, conf.level = 0.95)
```

Arguments

`score.indiv`	Numeric vector of individual scores.
`score.ref`	Numeric vector of reference scores (e.g., from a CSEM table).
`csem`	Numeric vector of CSEM values corresponding to `score.ref`.
`ci`	Logical. If `TRUE`, compute confidence intervals. Default `FALSE`.
`conf.level`	Numeric. Confidence level (default 0.95).

Value

A data frame with the same number of rows as `length(score.indiv)`, containing columns: `score.indiv`, `CSEM`, and if `ci = TRUE`, `lwr` and `upr`.

Examples

```
# Use data
library(psychTools)
data(ability)
data.ability <- ability[complete.cases(ability),]

# get CSEM deom Binomial model
res <- csemStrong(data.ability, score.type = "dich", nitems = 16)

# Get CSEM for scores 5, 8, 12 from a strong CSEM table
```

```
scoreCSEM(score.indiv = c(5, 8, 12),
          score.ref = res$CSEM$raw.score,
          csem = res$CSEM$csem.strong,
          ci = TRUE)
```

`stdCSEM` *Standardized Conditional Reliability (Raju et al., 2007)*

Description

Computes examinee-level (or score-level) reliability from conditional standard errors of measurement (CSEM) and the total observed score variance.

Usage

```
stdCSEM(csem, var_obs, na.rm = FALSE)
```

Arguments

`csem`	Numeric vector of conditional standard errors (e.g., from `csemMF` or `csemThorndike`).
`var_obs`	Numeric. Variance of the observed total scores.
`na.rm`	Logical. If TRUE, missing values in `csem` are removed before computation.

Details

The standardized conditional reliability for each score level (or examinee) is:

$$\rho = 1 - \frac{CSEM^2}{\sigma_X^2}$$

where σ_X^2 is the variance of observed scores in the sample. Values are truncated to the 0 to 1 interval.

Value

A numeric vector of the same length as `csem` (or shorter if `na.rm = TRUE`), with reliability estimates rounded to 3 decimal places.

References

Raju, N. S., Price, L. R., Oshima, T. C., & Nering, M. L. (2007). Standardized conditional SEM: A comparison of methods. *Educational and Psychological Measurement*, 67(6), 903-916.

Examples

```
# Sample of values
csem_vals <- c(2.0, 2.5, 3.0)
var_obs <- 25
stdCSEM(csem_vals, var_obs)

#From Strong true score model
library(psychTools)
data(ability)
data.ability <- ability[complete.cases(ability),]

strongCSEM.out <- csemStrong(score.type = "dich",
data = data.ability,
nitems = 16,
ci = TRUE,
summary = TRUE)

# Looking ouput
strongCSEM.out

# Standardized CSEM
stdCSEM(csem = strongCSEM.out$CSEM$csem.strong,
var_obs = strongCSEM.out$summary$value[3])
```

Index