Package 'contentvalidR'

September 28, 2026

Type Package

Title Tools for Substantive and Content Validity Pretesting

Version 0.4.0

Description Provides quantitative tools for substantive and content-oriented scale pretesting. Implements item-sort indices from Anderson and Gerbing (1991) <doi:10.1037/0021-9010.76.5.732>, exact item-sort inference following Howard and Melloy (2016) <doi:10.1007/s10869-015-9404-y>, empirical interpretation benchmarks from Colquitt et al. (2019) <doi:10.1037/apl0000406>, and the construct-rating procedure of Hinkin and Tracey (1999) <doi:10.1177/109442819922004> with HTC/HTD indices and repeated-measures item screening. The expert-panel workflow combines Aiken's V with score confidence intervals, Lawshe content validity ratios with exact inference, content validity indices with modified kappa and score intervals, item-objective congruence, and panel-level agreement using Krippendorff's alpha as described by Hayes and Krippendorff (2007) <doi:10.1080/19312450709336664>. Also provides judge and rater heterogeneity analysis following the generalizability-theory treatment of content-validity ratings in Crocker, Llabre and Miller (1988) <doi:10.1111/j.1745-3984.1988.tb00309.x>, content-domain coverage and expert-perceived content structure following Sireci and Geisinger (1992) <doi:10.1177/014662169201600102>, comparison across successive pretest rounds, and exact expert-panel planning. Where published methods compete, users choose among them through arguments with evidence-based defaults. User-facing workflows emphasize interpretable summaries and transparent review recommendations rather than isolated coefficients.

License GPL-3

Encoding UTF-8

Language en-US

Depends R (>= 4.0.0)

Imports stats

Suggests irr, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/JUhalt/contentvalidR,
https://juhalt.github.io/contentvalidR/

BugReports https://github.com/JUhalt/contentvalidR/issues

Config/Needs/website r-lib/pkgdown

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Joshua Uhalt [aut, cre]

Maintainer Joshua Uhalt <Josh.Uhalt@gmail.com>

Repository CRAN

Date/Publication 2026-09-28 11:10:02 UTC

Contents

`agreement_summary` *Agreement summary (auxiliary)*

Description

Computes Fleiss' kappa via 'irr' if available. This is an auxiliary compatibility helper, not part of the recommended contentvalidR workflows.

Usage

```
agreement_summary(ratings)
```

Arguments

`ratings`	matrix/data.frame: rows = items, cols = raters (nominal categories)

Value

When 'irr' is installed, the result of `irr::kappam.fleiss()`. Otherwise a list with `ok = FALSE` and a `message`, after a message explaining how to install 'irr'.

See Also

`panel_agreement()` for panel-level agreement with an evidence-based default coefficient and a bootstrap interval.

Examples

```
ratings <- data.frame(
  rater1 = c("A", "B", "A", "C"),
  rater2 = c("A", "B", "B", "C"),
  rater3 = c("A", "B", "A", "C")
)
if (requireNamespace("irr", quietly = TRUE)) {
  agreement_summary(ratings)
}
```

aikens_v *Aiken's V for expert content-relevance ratings*

Description

Computes Aiken's V per item for bounded ordinal expert ratings. By default, confidence intervals use the score method described by Penfield and Giacobbi (2004). Percentile bootstrap intervals remain available for compatibility and sensitivity analysis.

Usage

```
aikens_v(
  ratings,
  lo = 1,
  hi = 5,
  ci = c("score", "none", "bootstrap"),
  B = 500,
  alpha = 0.05,
  seed = NULL,
  na.rm = FALSE
)
```

Arguments

ratings	Matrix/data.frame with judges in rows and items in columns.
lo, hi	Numeric lower and upper bounds of the rating scale.
ci	Confidence-interval method: "score" (default), "bootstrap", or "none".
B	Number of bootstrap replicates when ci = "bootstrap".
alpha	Two-sided CI alpha level; .05 gives a 95% interval.
seed	Optional integer seed for bootstrap reproducibility.
na.rm	Logical. If FALSE (default), missing ratings are an error. If TRUE, item-specific effective judge counts are used.

Value

A data.frame with item, effective judge count N, number missing, Aiken's V, and (when requested) ci_low and ci_high.

References

Aiken, L. R. (1980). Content validity and reliability of single items or questionnaires. *Educational and Psychological Measurement, 40*(4), 955-959. doi:10.1177/001316448004000419

Penfield, R. D., & Giacobbi, P. R., Jr. (2004). Applying a score confidence interval to Aiken's item content-relevance index. *Measurement in Physical Education and Exercise Science, 8*(4), 213-225. doi:10.1207/S15327841MPEE0804_3

Examples

```
R <- matrix(c(4,4,3,4, 4,3,4,4, 3,3,4,4), nrow = 4)
colnames(R) <- c("Item1", "Item2", "Item3")
aikens_v(R, lo = 1, hi = 4)
```

`anova_content` *Hinkin-Tracey ANOVA content test*

Description

For each item, evaluates whether definitional-correspondence ratings differ across construct definitions and whether the intended construct is rated higher than every orbiting construct.

The Hinkin and Tracey (1999) rating task is ordinarily a **within-judge** design: the same judge rates an item against multiple construct definitions. For that design, `anova_content()` uses a one-way repeated-measures ANOVA on judges with complete ratings for the item's construct set, followed by one-sided paired planned contrasts of the target against each orbiting construct. A between-judge path is retained for genuinely independent rating designs, but it is not the recommended Hinkin-Tracey protocol.

The repeated-measures output includes the conventional omnibus F/p and a Greenhouse-Geisser epsilon/corrected p-value. With more than two construct definitions, the corrected p-value is the safer default for omnibus screening when sphericity may not hold. Planned target-versus-orbiting contrasts provide the more direct item-level evidence.

Usage

```
anova_content(
  ratings,
  item_col = "item",
  rater_col = "rater",
  construct_col = "construct",
  rating_col = "rating",
  target_map = NULL,
  posthoc = NULL,
  alpha = 0.05,
  target_col = "target_construct",
  design = c("auto", "within", "between"),
  adjust = c("none", "holm")
)
```

Arguments

`ratings`	A long-format data.frame with item, rater, construct, and numeric rating columns.
`item_col`, `rater_col`, `construct_col`, `rating_col`	Column names.

target_map	Optional named character vector/list mapping item to target. If neither a map nor target_col is available, omnibus tests are still returned but target-versus-orbiting contrasts are NA.
posthoc	Deprecated compatibility argument. Tukey/Duncan post-hoc testing is no longer used because the Hinkin-Tracey question is directly represented by planned target-versus-orbiting contrasts.
alpha	Significance level for the omnibus test and planned contrasts.
target_col	Target column used when target_map is NULL.
design	One of "auto", "within", or "between". "auto" identifies the design itemwise from whether judges provide ratings for multiple construct definitions.
adjust	Multiplicity adjustment for the target-versus-orbiting planned contrast p values. Default "none" reproduces the planned-comparison logic commonly used with the Hinkin-Tracey procedure; "holm" is a conservative option.

Value

A data.frame with one row per item, including the omnibus F, raw p, Greenhouse-Geisser epsilon/corrected degrees of freedom and p-value for within-judge designs, partial eta-squared, and planned-contrast diagnostics. The full planned-contrast table is stored in attr(result, "contrasts"). posthoc_pass is retained as an alias of contrast_pass for backward compatibility.

References

Hinkin, T. R., & Tracey, J. B. (1999). An analysis of variance approach to content validation. *Organizational Research Methods, 2*(2), 175-186. doi:10.1177/109442819922004

Colquitt, J. A., Baer, M. D., Long, D. M., & Halvorsen-Ganepola, M. D. K. (2014). Scale indicators of social exchange relationships: A comparison of relative content validity. *Journal of Applied Psychology, 99*(4), 599-618. doi:10.1037/a0036374

Examples

```
set.seed(1)
d <- expand.grid(item = c("I1", "I2"), rater = 1:12,
                 construct = c("A", "B", "C"))
d$target_construct <- ifelse(d$item == "I1", "A", "B")
d$rating <- ifelse(d$construct == d$target_construct,
                   rnorm(nrow(d), 4.5, .4), rnorm(nrow(d), 2.3, .5))
anova_content(d)
```

---

as.data.frame.contentvalid_workflow

Extract workflow results as a plain data frame

Description

Returns a fitted workflow's results as an ordinary data frame, so results can be filtered, joined, or written out without scraping printed output.

A `workflow` column is prepended so that tables from several analyses can be stacked and stay identifiable.

Usage

```
## S3 method for class 'contentvalid_workflow'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  component = c("results", "scale_summary"),
  include_interpretation = TRUE,
  ...
)
```

Arguments

`x`	A fitted `contentvalid_workflow` object.
`row.names`, `optional`	Present for compatibility with the generic.
`component`	Which component to return: `"results"` (the default, one row per unit of analysis) or `"scale_summary"`.
`include_interpretation`	Keep the per-unit interpretation text. It is informative but long, so set `FALSE` for compact tables.
`...`	Ignored.

Value

A data frame.

Filtering is your decision, not the package's

There is deliberately no helper that returns "the items that passed." Selecting on `status == "Supported"` is a substantive decision that should appear in your own code where a reader can see it, and `Review` never means an item must be dropped. Keeping the filter explicit keeps that judgment visible in the analysis script and in the manuscript.

Examples

```
sorts <- read.csv(
  system.file("extdata", "sort_example.csv", package = "contentvalidR"),
  stringsAsFactors = FALSE
)
fit <- sort_validity(sorts)
```

```
head(as.data.frame(fit, include_interpretation = FALSE))
as.data.frame(fit, component = "scale_summary")
```

`colquitt_benchmarks` *Colquitt et al. (2019) empirical content-validation benchmarks*

Description

Returns the empirical interpretation bands proposed by Colquitt et al. (2019) for Psa, Csv, HTC, or HTD. The benchmarks were created from scale-level averages for 112 scales and are percentile-based norms, not universal psychometric cutoffs.

If `orbiting_r` is supplied, the correlation-conditional benchmark set is selected. Otherwise the overall, non-correlation-normed criteria are used.

The published table contains a few rounded boundary overlaps/gaps. This implementation treats each printed lower bound as the start of its category and assigns categories from strongest to weakest, yielding deterministic interpretation at rounded boundaries.

Usage

```
colquitt_benchmarks(
  statistic = c("psa", "csv", "htc", "htd"),
  orbiting_r = NULL
)
```

Arguments

`statistic`	One of `"psa"`, `"csv"`, `"htc"`, or `"htd"`.
`orbiting_r`	Optional average correlation between the focal scale and its orbiting scales.

Value

A data.frame describing the selected benchmark set and its lower cutpoints.

References

Colquitt, J. A., Sabey, T. B., Rodell, J. B., & Hill, E. T. (2019). Content validation guidelines: Evaluation criteria for definitional correspondence and definitional distinctiveness. *Journal of Applied Psychology, 104*(10), 1243-1265. doi:10.1037/apl0000406

Examples

```
colquitt_benchmarks("psa")
colquitt_benchmarks("csv", orbiting_r = .40)
```

`compare_rounds` *Compare content-validity evidence across pretest rounds*

Description

Compares two or more fitted workflow objects from successive rounds of the same pretest, reporting which items changed status, which held steady, and which entered or left the item set.

Scale development is iterative: items get revised and re-tested. The risk in reporting that process is attributing a status change to improved items when it actually came from a changed decision rule, a different panel size, or a different criterion. This function makes that distinction visible by comparing the `settings` of each round alongside its results, and flagging rounds whose analysis settings differ.

Usage

```
compare_rounds(..., labels = NULL)
```

Arguments

`...`	Two or more fitted workflow objects, in round order. All must come from the same workflow, since status labels from different workflows rest on different criteria and are not comparable.
`labels`	Optional round labels. Defaults to `Round 1`, `Round 2`, and so on, or to the names supplied in `....`

Value

An object of class `contentvalid_rounds`, a list containing:

transitions One row per unit, with its status in each round and the direction of any change.

summary Counts of stable, improved, weakened, added, and removed units for each consecutive pair of rounds.

settings_changes Analysis settings that differ between consecutive rounds, which is the audit trail for whether a status change can be read as an evidence change at all.

comparable `FALSE` when any consecutive pair differs in settings.

Reading a comparison

A status change means the evidence crossed a criterion, not that an item improved by a measurable amount. An item can move from `Review` to `Supported` on a small change in one judge's rating if it was sitting near the boundary. Read the transitions together with the underlying index values in each round's own results.

When `comparable` is `FALSE`, the rounds were analyzed under different rules, and a status change may reflect only that. Re-analyze the earlier round under the current settings before reporting a change as progress.

See Also

reproducibility_phi() for agreement between two independent judge samples analyzed under identical settings.

Examples

```
round1 <- data.frame(
  item = rep(c("I1", "I2"), each = 6),
  rater = rep(1:6, times = 2),
  assigned_construct = c(rep("A", 5), "B", rep("A", 3), rep("B", 3)),
  target_construct = "A",
  stringsAsFactors = FALSE
)
round2 <- round1
round2$assigned_construct <- c(rep("A", 6), rep("A", 5), "B")
compare_rounds(sort_validity(round1), sort_validity(round2))
```

---

compute_csv *Substantive Validity Coefficient (Csv)*

Description

For each item, computes Anderson and Gerbing's (1991) substantive-validity coefficient:

$$C_{sv} = (n_c - n_o)/N,$$

where n_c is the number of non-missing assignments to the intended construct, n_o is the largest number of assignments to any one non-target construct, and N is the number of non-missing assignments.

Missing assignments are excluded itemwise and reported in n_missing.

Usage

```
compute_csv(
  assignments,
  item_col = "item",
  rater_col = "rater",
  assigned_col = "assigned_construct",
  target_col = "target_construct"
)
```

Arguments

assignments	A data.frame containing item-sort responses.
item_col, rater_col, assigned_col, target_col	Column names for the item, rater, assigned construct, and intended target construct.

Value

A data.frame with one row per item and columns `item`, `target`, `n_total`, `n`, `n_missing`, `n_target`, `competitor`, `n_other_max`, and `csv`.

References

Anderson, J. C., & Gerbing, D. W. (1991). Predicting the performance of measures in a confirmatory factor analysis with a pretest assessment of their substantive validities. *Journal of Applied Psychology, 76*(5), 732-740. doi:10.1037/00219010.76.5.732

Examples

```
df <- data.frame(
  item = rep(c("I1", "I2"), each = 4),
  rater = rep(1:4, 2),
  assigned_construct = c("A", "A", "A", "B", "B", "B", "B", "B"),
  target_construct = rep("A", 8)
)
compute_csv(df)
```

`compute_psa` *Proportion of Substantive Agreement (Psa)*

Description

For each item, computes the proportion of non-missing item-sort responses assigned to the item's intended (target) construct. This is Anderson and Gerbing's (1991) proportion of substantive agreement, P_{sa}.

Missing assignments are excluded itemwise and reported in `n_missing` so that the effective denominator is transparent. Psa is a proportion of a finite set of judges, so an interval is reported alongside it; see `ci`.

Usage

```
compute_psa(
  assignments,
  item_col = "item",
  rater_col = "rater",
  assigned_col = "assigned_construct",
  target_col = "target_construct",
  ci = c("wilson", "agresti_coull", "exact", "none"),
  alpha = 0.05
)
```

Arguments

`assignments`	A data.frame containing item-sort responses.
`item_col`, `rater_col`, `assigned_col`, `target_col`	Column names for the item, rater, assigned construct, and intended target construct.
`ci`	Interval method for Psa: `"wilson"` (default), `"agresti_coull"`, `"exact"`, or `"none"`. The methods and the evidence for each are described under `ci` in `cvi()`.
`alpha`	Two-sided alpha level for the interval; `0.05` gives a 95% interval.

Value

A data.frame with one row per item and columns `item`, `target`, `n_total`, `n`, `n_missing`, `n_target`, `psa`, `psa_low`, and `psa_high`.

References

Anderson, J. C., & Gerbing, D. W. (1991). Predicting the performance of measures in a confirmatory factor analysis with a pretest assessment of their substantive validities. *Journal of Applied Psychology, 76*(5), 732-740. doi:10.1037/00219010.76.5.732

Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association, 22*(158), 209-212. doi:10.1080/01621459.1927.10502953

Newcombe, R. G. (1998). Two-sided confidence intervals for the single proportion: Comparison of seven methods. *Statistics in Medicine, 17*(8), 857-872.

Agresti, A., & Coull, B. A. (1998). Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician, 52*(2), 119-126. doi:10.1080/00031305.1998.10480550

Clopper, C. J., & Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika, 26*(4), 404-413. doi:10.1093/biomet/26.4.404

See Also

`cvi()` for the same interval methods applied to I-CVI.

Examples

```
df <- data.frame(
  item = rep(c("I1", "I2"), each = 4),
  rater = rep(1:4, 2),
  assigned_construct = c("A", "A", "A", "B", "B", "A", "B", "B"),
  target_construct = rep(c("A", "B"), each = 4)
)
compute_psa(df)
compute_psa(df, ci = "exact")
```

`contentvalid_glossary` *Glossary of contentvalidR indices and status terms*

Description

Plain-language definitions of every abbreviated quantity the package reports, and of the status labels shared by all flagship workflows.

The same definitions are printed beneath workflow output, so what you read here is what appears alongside your results. Set `options(contentvalidR.show_key = FALSE)` to suppress those inline keys once the terms are familiar.

Usage

```
contentvalid_glossary(workflow = NULL)
```

Arguments

`workflow`	Optionally restrict to one workflow: `"item-sort"`, `"construct-rating"`, `"expert-panel"`, `"judge-heterogeneity"`, or `"domain-coverage"`.

Value

An object of class `contentvalid_glossary`: a data frame of `term`, `workflow`, `label`, `definition`, and `range`, carrying the status definitions as the `"statuses"` attribute.

A note on benchmark labels

Strength labels such as `Strong` or `Weak` from `interpret_colquitt()` are percentile positions relative to scales published in the measurement literature. They are not absolute judgments, and they are not comparable across indices: HTC and HTD sit on different scales with different typical values, so an HTC of 0.83 can be labeled `Weak` in the same analysis where an HTD of 0.44 is labeled `Very Strong`. Compare each index against its own benchmark, never against another index's number.

See Also

`interpret_colquitt()` for the benchmark bands themselves.

Examples

```
contentvalid_glossary()
contentvalid_glossary("item-sort")
```

content_handoff *Carry content-validity decisions into empirical validation*

Description

Packages the item decisions from a finished content-validity workflow so they can be carried into an empirical scale-development workflow without retyping item names or losing the record of why each item was kept.

The result holds the item names that survived content review, the construct each belongs to where the design defines one, a per-item evidence table, the statistics behind each decision, and the provenance of the analysis. It is plain data, so a downstream package can read it without contentvalidR being installed.

Usage

```
content_handoff(fit, keep = "Supported", round = 1)
```

Arguments

fit	A fitted contentvalid_sort, contentvalid_rating, or contentvalid_expert object.
keep	Statuses that travel forward, defaulting to "Supported". Any of "Supported", "Review", "Insufficient data", or "Descriptive only".
round	Pretest round this analysis represents. One fit is one round, so this defaults to 1 and matters only when stacking rounds by hand.

Details

Item-level workflows are accepted: sort_validity(), rating_validity(), and expert_validity(). judge_validity() and domain_validity() are refused, because their rows are judges and blueprint cells rather than items, so there is no item set to carry forward.

Items that do not meet keep are not dropped from the record. They stay in item_evidence with carried = FALSE, so a reader can see what was held back and why. Review is not deletion.

Value

An object of class contentvalid_handoff, cv_handoff, and list, as described under "Object shape".

Object shape (schema version 1)

The object has class c("contentvalid_handoff", "cv_handoff", "list"). A consumer matches on "cv_handoff" and reads these fields:

items character vector of the carried item names, that is item_evidence$item[item_evidence$carried], unique and in results order.

`scales` named list mapping each construct to its carried items, or `NULL` when the design has no construct mapping. Expert relevance and essentiality rate a single item set with no construct column, so they produce `NULL`. Membership is one to one.

`item_evidence` data frame with one row per reviewed item: `item`, `scale` (`NA` without a construct mapping), `carried`, `status`, `recommendation`, `n_judges`, `rule`, and `round`.

`item_statistics` data frame, one row per item per statistic: `item`, `statistic`, `value`, and `criterion` (`NA` when the method sets no explicit criterion).

`provenance` list with `schema_version`, `package`, `package_version`, `workflow`, `mode`, `keep`, `method`, `citation`, `settings`, `design`, and `created`.

This shape is agreed with the `nomologR` package, which consumes it in `nomo_screen()` and `nomo_run()`. Neither package depends on the other.

What a handoff does and does not establish

Surviving content review is evidence about relevance, representation, and expert judgment. It does not establish that an item will behave well empirically. An item can be clearly relevant and still correlate poorly with its construct or load on an unintended factor. That is what the downstream empirical analysis tests, which is why the item set travels with its evidence rather than as a bare list of names.

See Also

`as.data.frame.contentvalid_workflow()` for the full results table, and `content_report()` for manuscript tables.

Examples

```
relevance <- matrix(
  c(4,4,4,3, 4,4,3,4, 3,4,4,4, 2,2,1,2),
  nrow = 4,
  dimnames = list(NULL, paste0("Item", 1:4))
)
fit <- expert_validity(relevance, mode = "relevance", lo = 1, hi = 4,
                       agreement = "none")
handoff <- content_handoff(fit)
handoff
handoff$items
handoff$item_evidence
handoff$item_statistics

# Carry items flagged for review as well, when the study protocol says so.
content_handoff(fit, keep = c("Supported", "Review"))$items
```

`content_report` *Build a manuscript-ready results table*

Description

Formats a fitted workflow's results as a compact table suitable for pasting into a manuscript or a Quarto or R Markdown document, either as a data frame or as a Markdown table.

Markdown output is generated directly, so no reporting package is required to use it. Nothing in the core analysis depends on one.

Usage

```
content_report(
  x,
  digits = 2,
  format = c("data.frame", "markdown"),
  include = c("all", "flagged"),
  caption = NULL
)
```

Arguments

`x`	A fitted `contentvalid_workflow` object.
`digits`	Digits for rounding numeric columns.
`format`	`"data.frame"` (default) or `"markdown"`.
`include`	`"all"` (default) or `"flagged"`, which keeps only units whose status is not `Supported`.
`caption`	Optional caption line placed above a Markdown table.

Value

A data frame, or a character vector of Markdown lines when `format = "markdown"`. The character vector carries the analysis provenance as its `"settings"` attribute.

Reporting the decision rules

A results table alone is not a reproducible report. The thresholds that produced each status live in the fitted object's `settings`, and are attached to Markdown output as an attribute so they travel with the table. Report them alongside it: two analyses of identical data can disagree entirely because one used a different criterion.

See Also

`as.data.frame.contentvalid_workflow()` for the untrimmed table, and `compare_rounds()` for reporting change across pretest rounds.

Examples

```
sorts <- read.csv(
  system.file("extdata", "sort_example.csv", package = "contentvalidR"),
  stringsAsFactors = FALSE
)
fit <- sort_validity(sorts)
content_report(fit)
cat(content_report(fit, format = "markdown", include = "flagged"), sep = "\n")
```

---

content_structure *Item-similarity structure of a content domain*

Description

Analyzes whether subject-matter experts perceive items as grouping the way a test blueprint says they should, using the multidimensional scaling and cluster analysis procedure of Sireci and Geisinger (1992, 1995).

Experts rate how similar each pair of items is. Those similarities are scaled into a low-dimensional content map and clustered. If the blueprint describes the domain as experts actually see it, the recovered clusters should correspond to the blueprint's cells. Agreement is quantified with the adjusted Rand index, which is corrected for chance so that a value near 0 means no better than random correspondence.

This is evidence about perceived content *structure*. It is not evidence that the items cover the domain: see `domain_validity()` for coverage.

Usage

```
content_structure(
  similarity,
  membership = NULL,
  k = NULL,
  dims = 2,
  max_dims = 5,
  similarity_is_distance = FALSE
)
```

Arguments

`similarity`	A square, symmetric item-by-item matrix of expert similarity ratings, or a distance matrix when `similarity_is_distance` is `TRUE`. Similarities are converted to distances as `max(similarity) - similarity`.
`membership`	Optional blueprint cell for each item, as a vector in the same order as the rows of `similarity`, or named by item. When supplied, the recovered clustering is compared against it.
`k`	Number of clusters to extract. Defaults to the number of distinct blueprint cells, or 2 when no blueprint is supplied.

`dims` Number of multidimensional scaling dimensions to retain.

`max_dims` Largest dimensionality reported in the fit table.

`similarity_is_distance`
Set `TRUE` when `similarity` already holds distances rather than similarities.

Value

An object of class `contentvalid_structure`, a list containing the MDS `coordinates`, `clusters`, the `fit` table across dimensionalities, the `stress` and `gof` of the retained solution, the `adjusted_rand` index and `cross_tab` against the blueprint, `settings`, `design`, `status`, and an `interpretation`.

Dimensionality

The retained dimensionality is reported rather than chosen silently. The `fit` table gives Kruskal stress-1 for every dimensionality up to `max_dims`, with the conventional descriptive labels. Those labels are long-standing conventions for describing fit, not thresholds that decide how many dimensions a content domain has. Substantive interpretability of the dimensions should drive that choice.

References

Sireci, S. G., & Geisinger, K. F. (1992). Analyzing test content using cluster analysis and multidimensional scaling. *Applied Psychological Measurement, 16*(1), 17-31. doi:10.1177/014662169201600102

Sireci, S. G., & Geisinger, K. F. (1995). Using subject-matter experts to assess content representation: An MDS analysis. *Applied Psychological Measurement, 19*(3), 241-255. doi:10.1177/014662169501900303

Sireci, S. G. (1998). The construct of content validity. *Social Indicators Research, 45*(1-3), 83-117. doi:10.1023/A:1006985528729

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*(1), 193-218. doi:10.1007/BF01908075

See Also

`similarity_from_sort()` to derive similarities from an item-sort task, and `domain_validity()` for the combined coverage-and-structure workflow.

Examples

```
items <- paste0("I", 1:6)
blueprint <- c(rep("Autonomy", 3), rep("Competence", 3))
sim <- matrix(1, 6, 6, dimnames = list(items, items))
sim[1:3, 1:3] <- 5
sim[4:6, 4:6] <- 5
diag(sim) <- 5
content_structure(sim, membership = blueprint)
```

`csv_binom_test` *Exact item-sort significance test*

Description

Tests whether the number of assignments to an item's intended construct exceeds the count expected under a binomial chance model. With the default `p0 = 0.5`, this implements the Howard and Melloy (2016) retention test for item-sort tasks by testing the target-assignment count directly. Unlike the legacy critical-Csv procedure, the count-based test remains applicable when respondents choose among more than two construct alternatives.

The function name is retained for backward compatibility even though the inferential test is performed on `n_c`, not on the observed Csv value.

Usage

```
csv_binom_test(n_c, N, p0 = 0.5, alpha = 0.05)
```

Arguments

`n_c`	Integer; number of non-missing assignments to the target construct.
`N`	Integer; total number of non-missing assignments for the item.
`p0`	Null target-assignment probability. Default `0.5`, following Howard and Melloy (2016).
`alpha`	Significance level. Default `0.05`.

Value

A list containing the exact p-value, observed target proportion, one-sided confidence interval, the minimum critical target count, a logical `passes_chance` flag, a backward-compatible `decision` label, and a plain-language `interpretation`.

References

Howard, M. C., & Melloy, R. C. (2016). Evaluating item-sort task methods: The presentation of a new statistical significance formula and methodological best practices. *Journal of Business and Psychology, 31*(1), 173-186. doi:10.1007/s108690159404y

Examples

```
csv_binom_test(n_c = 15, N = 20)
csv_binom_test(n_c = 14, N = 20)
```

cvi *Content Validity Index (CVI)*

Description

Computes item-level Content Validity Index (I-CVI), scale-level average CVI (S-CVI/Ave), universal-agreement CVI (S-CVI/UA), and the modified kappa described by Polit, Beck, and Owen (2007).

For each item, modified kappa adjusts I-CVI for chance agreement using the probability of observing exactly `A` agreements among `N` judges:

$$P_c = \binom{N}{A}(0.5)^N$$

and

$$k^* = (I_C VI - P_c)/(1 - P_c).$$

I-CVI is a proportion of what is usually a small panel, so an interval is reported alongside it. The interval method is selectable; see `ci`.

Usage

```
cvi(
  binary,
  na.rm = FALSE,
  ci = c("wilson", "agresti_coull", "exact", "none"),
  alpha = 0.05
)
```

Arguments

`binary`	Matrix/data.frame with judges in rows and items in columns, coded `1 = relevant` and `0 = not relevant`.
`na.rm`	Logical. If `FALSE` (default), missing ratings are an error. If `TRUE`, missing ratings are removed itemwise and each item's effective judge count is reported in `N`.
`ci`	Interval method for I-CVI: • `"wilson"` (default): the Wilson (1927) score interval. Newcombe (1998) compared seven methods and recommends score intervals over the Wald interval. • `"agresti_coull"`: the adjusted Wald interval of Agresti and Coull (1998), which they show performs well even in small samples. Limits are clipped so they stay between 0 and 1. • `"exact"`: the Clopper and Pearson (1934) interval. It is conservative: Agresti and Coull (1998) show its coverage runs above the nominal level. • `"none"`: no interval is computed, and the interval columns are `NA`.
`alpha`	Two-sided alpha level for the interval; `0.05` gives a 95% interval.

Value

A classed list with:

- `item_level`: item, A, N, I_CVI, I_CVI_low, I_CVI_high, Pc, kappa_mod
- `scale_level`: S_CVI_Ave and S_CVI_UA
- `ci` and `alpha`: the interval settings used

References

Polit, D. F., Beck, C. T., & Owen, S. V. (2007). Is the CVI an acceptable indicator of content validity? Appraisal and recommendations. *Research in Nursing & Health, 30*(4), 459-467. doi:10.1002/nur.20199

Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association, 22*(158), 209-212. doi:10.1080/01621459.1927.10502953

Newcombe, R. G. (1998). Two-sided confidence intervals for the single proportion: Comparison of seven methods. *Statistics in Medicine, 17*(8), 857-872.

Agresti, A., & Coull, B. A. (1998). Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician, 52*(2), 119-126. doi:10.1080/00031305.1998.10480550

Clopper, C. J., & Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika, 26*(4), 404-413. doi:10.1093/biomet/26.4.404

Examples

```
M <- matrix(
  c(1,1,1,1, 1,1,1,0, 1,1,0,0),
  nrow = 4,
  dimnames = list(NULL, c("Item1", "Item2", "Item3"))
)
cvi(M)
cvi(M, ci = "exact")
```

`cvr` *Lawshe's Content Validity Ratio (CVR)*

Description

Computes Lawshe's CVR and exact one-sided binomial inference following the critical-value logic revisited by Ayre and Scally (2014). Input may be either counts of experts marking each item essential or a judge-by-item 0/1 matrix.

Usage

```
cvr(essential, N = NULL, alpha = 0.05, na.rm = FALSE, item_names = NULL)
```

Arguments

essential	Numeric/integer vector of essential counts, or a matrix/data frame with judges in rows, items in columns, coded 1 = essential and 0 = not essential.
N	Panel size. Required for count-vector input. May be a scalar or a vector matching essential. Ignored for matrix input, where effective N is calculated itemwise.
alpha	One-sided exact alpha level. Default .05.
na.rm	Logical; for matrix input, permit itemwise missing ratings.
item_names	Optional item names for count-vector input.

Value

A data.frame containing item, ne, effective N, CVR, exact p-value, critical essential count/CVR, and pass.

References

Lawshe, C. H. (1975). A quantitative approach to content validity. *Personnel Psychology, 28*(4), 563-575. doi:10.1111/j.17446570.1975.tb01393.x

Ayre, C., & Scally, A. J. (2014). Critical values for Lawshe's content validity ratio: Revisiting the original methods of calculation. *Measurement and Evaluation in Counseling and Development, 47*(1), 79-86. doi:10.1177/0748175613513808

Examples

```
cvr(essential = c(8, 10, 5), N = 12)
```

---

domain_validity *Analyze content-domain coverage and structure*

Description

Answers two questions that item-level relevance indices cannot: whether the item set actually spans the intended content domain, and whether experts perceive the items as grouping the way the blueprint says they should.

Coverage is assessed against a blueprint, or table of specifications: the cells of the domain the instrument is meant to represent. Cells with no items, or too few, are content gaps that no amount of item-level relevance evidence will reveal, because an item can only be rated if it exists.

Structure is assessed with the multidimensional scaling and cluster analysis procedure of Sireci and Geisinger (1992), and is run when expert similarity data is supplied. See content_structure().

Usage

```
domain_validity(
  assignments,
  item_col = "item",
  cell_col = "cell",
  facet_col = NULL,
  domain = NULL,
  min_items = 2,
  over_factor = 2,
  targets = NULL,
  similarity = NULL,
  ...
)
```

Arguments

`assignments`	A data frame mapping items to blueprint cells.
`item_col`	Column naming each item.
`cell_col`	Column naming each item's blueprint cell, typically the construct or content area.
`facet_col`	Optional second column. When supplied, cells are the crossing of `cell_col` and `facet_col`, as in a construct-by-facet table of specifications.
`domain`	Optional character vector of every cell the blueprint intends to cover. Supplying it is what makes **empty** cells detectable; without it only the cells that already contain items can be reported.
`min_items`	Fewest items a cell may hold before it is flagged as thinly covered.
`over_factor`	A cell holding more than this multiple of its expected share is flagged as over-represented. This is an attention-drawing heuristic, not a standard.
`targets`	Optional named numeric vector giving the intended number of items per cell. When supplied, expected shares come from it rather than from an assumption of equal cells.
`similarity`	Optional square item-by-item expert similarity matrix. When supplied, the content-structure analysis is run and reported alongside coverage.
`...`	Further arguments passed to `content_structure()`.

Value

An object of class `contentvalid_domain` and `contentvalid_workflow`. `results` has **one row per blueprint cell**. `details$structure` holds the content-structure analysis when similarity data was supplied.

What coverage evidence can and cannot establish

A fully covered blueprint shows that items exist for every intended cell. It does not show that those items are good ones, that the blueprint itself is the right description of the domain, or that the cells are equally important. Coverage is evidence about the item set's reach, and is properly read alongside item-level relevance evidence and expert judgment about the blueprint itself.

References

Sireci, S. G. (1998). The construct of content validity. *Social Indicators Research, 45*(1-3), 83-117. doi:10.1023/A:1006985528729

Sireci, S. G., & Geisinger, K. F. (1992). Analyzing test content using cluster analysis and multidimensional scaling. *Applied Psychological Measurement, 16*(1), 17-31. doi:10.1177/014662169201600102

Rovinelli, R. J., & Hambleton, R. K. (1977). On the use of content specialists in the assessment of criterion-referenced test item validity. *Dutch Journal of Educational Research, 2*, 49-60.

See Also

`content_structure()`, `similarity_from_sort()`, `ioc()`.

Examples

```
assignments <- data.frame(
  item = paste0("I", 1:7),
  construct = c("Autonomy", "Autonomy", "Autonomy", "Autonomy",
                "Competence", "Competence", "Relatedness")
)
domain_validity(
  assignments,
  cell_col = "construct",
  domain = c("Autonomy", "Competence", "Relatedness", "Belonging")
)
```

---

expert_power *Plan an expert panel against an explicit decision criterion*

Description

Reports the probability that an item will clear its expert-panel criterion, given a panel size and an assumed probability that a single expert endorses the item.

This replaces advice of the form "use six experts" with a question that has an answer: *if an expert endorses this item with probability* `prob`*, how often will a panel of this size actually clear the criterion?* Nothing here recommends a panel size. It reports the consequences of the sizes you ask about, so the choice stays yours and stays documented.

Usage

```
expert_power(
  n_experts = 3:12,
  prob = c(0.7, 0.8, 0.9),
  criterion = c("cvi", "cvr"),
  alpha = 0.05,
  response_rate = 1
)
```

Arguments

`n_experts`	Panel sizes to evaluate.
`prob`	Probability that one expert endorses the item, as relevant (`criterion = "cvi"`) or essential (`criterion = "cvr"`). Values well below 0.5 describe items the panel largely rejects.
`criterion`	`"cvi"` uses the common panel-size I-CVI guideline: 1.00 for three to five experts, 0.78 for six or more. `"cvr"` uses the exact Lawshe critical count at level `alpha`.
`alpha`	Significance level for the CVR criterion. Ignored for CVI.
`response_rate`	Expected proportion of invited experts who return usable ratings. When below 1, `n_experts` is treated as the number invited and the realized panel size is averaged over, so the reported probability accounts for both a smaller panel and the criterion that a smaller panel triggers.

Value

An object of class `contentvalid_expert_power`: a list whose `results` data frame holds one row per panel size and probability, with the required endorsement count and the probability of clearing.

Why the curve is not always smooth

The I-CVI criterion is a step function of panel size: it is 1.00 up to five experts and 0.78 from six. Adding a sixth expert relaxes the criterion and can raise the clearing probability sharply, while adding a fourth or fifth expert under unanimity makes clearing *harder*. A planning curve that rose smoothly with panel size would be hiding this, so it is reported as it is.

References

Lynn, M. R. (1986). Determination and quantification of content validity. *Nursing Research, 35*(6), 382-385.

Polit, D. F., Beck, C. T., & Owen, S. V. (2007). Is the CVI an acceptable indicator of content validity? *Research in Nursing & Health, 30*(4), 459-467. doi:10.1002/nur.20199

Ayre, C., & Scally, A. J. (2014). Critical values for Lawshe's content validity ratio. *Measurement and Evaluation in Counseling and Development, 47*(1), 79-86. doi:10.1177/0748175613513808

See Also

`sort_power()` for item-sort planning, and `gtheory_content()` whose decision study plans panel size against a generalizability target.

Examples

```
expert_power(n_experts = 3:10, prob = c(0.8, 0.9))
expert_power(n_experts = c(5, 10, 15), prob = 0.75, criterion = "cvr")
```

`expert_validity` *Analyze expert-panel content-validity evidence*

Description

Provides a user-facing workflow for three common expert-panel tasks:

- `mode = "relevance"`: bounded ordinal relevance ratings, combining Aiken's V (with Penfield-Giacobbi score intervals), CVI/modified kappa, and a panel-level agreement coefficient.
- `mode = "essentiality"`: Lawshe CVR with exact binomial critical values.
- `mode = "congruence"`: Rovinelli-Hambleton item-objective congruence.

Quantitative results are presented as evidence for item review rather than as a substitute for expert comments, construct coverage, comprehensibility, or other parts of a content-validity argument.

Usage

```
expert_validity(
  data,
  mode = c("relevance", "essentiality", "congruence"),
  lo = 1,
  hi = 4,
  relevance_cut = NULL,
  N = NULL,
  alpha = 0.05,
  na.rm = FALSE,
  target_col = "target_objective",
  proportion_ci = c("wilson", "agresti_coull", "exact", "none"),
  agreement = c("krippendorff", "ac1", "none"),
  agreement_level = c("ordinal", "nominal", "interval"),
  agreement_B = 1000,
  seed = NULL
)
```

Arguments

`data`	Ratings data. For relevance, a judge-by-item numeric matrix/data frame. For essentiality, either a judge-by-item 0/1 matrix/data frame or a vector of essential counts. For congruence, a long data frame accepted by `ioc()`.
`mode`	One of `"relevance"`, `"essentiality"`, or `"congruence"`.
`lo, hi`	Rating-scale bounds for relevance mode.
`relevance_cut`	Lowest rating treated as relevant for CVI. Defaults to `hi - 1`, e.g., 3 on a 1-4 scale or 4 on a 1-5 scale.
`N`	Panel size for essential-count vector input.
`alpha`	Inferential/CI alpha level.

na.rm	Permit itemwise/cellwise missing ratings where supported.
target_col	In congruence mode, optional column identifying each item's intended objective. If absent, IOC cells are returned descriptively.
proportion_ci	Interval method for I-CVI in relevance mode: "wilson" (default), "agresti_coull", "exact", or "none". The interval uses the same alpha as Aiken's V. See ci in cvi() for the methods and the evidence for each.
agreement	Panel-level agreement coefficient for relevance mode: "krippendorff" (default), "ac1", or "none". Krippendorff's alpha uses the relevance ratings at agreement_level; Gwet's AC1 uses the relevant/not-relevant decision. See panel_agreement() for the evidence behind each, including why AC1 is never the default. Panels with fewer than two experts or two items report no agreement coefficient.
agreement_level	Measurement level for Krippendorff's alpha: "ordinal" (default), "nominal", or "interval". Ignored for AC1.
agreement_B	Bootstrap resamples for the agreement interval; 0 skips the interval.
seed	Optional seed that makes the agreement interval reproducible.

Value

An object of class contentvalid_expert and contentvalid_workflow. All flagship workflow objects expose the common components results, scale_summary, settings, design, and details. The historical top-level scale component is retained as a compatibility alias for scale_summary. Results include a standardized status field while retaining mode-specific recommendation wording. In relevance mode, scale_summary also holds agreement, agreement_low, and agreement_high, and details$agreement holds the full panel_agreement() result.

References

Penfield, R. D., & Giacobbi, P. R., Jr. (2004). Applying a score confidence interval to Aiken's item content-relevance index. *Measurement in Physical Education and Exercise Science, 8*(4), 213-225. doi:10.1207/s15327841mpee0804_3

Ayre, C., & Scally, A. J. (2014). Critical values for Lawshe's content validity ratio: Revisiting the original methods of calculation. *Measurement and Evaluation in Counseling and Development, 47*(1), 79-86. doi:10.1177/0748175613513808

Polit, D. F., Beck, C. T., & Owen, S. V. (2007). Is the CVI an acceptable indicator of content validity? *Research in Nursing & Health, 30*(4), 459-467. doi:10.1002/nur.20199

Hayes, A. F., & Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures, 1*(1), 77-89. doi:10.1080/19312450709336664

Zapf, A., Castell, S., Morawietz, L., & Karch, A. (2016). Measuring inter-rater reliability for nominal data: Which coefficients and confidence intervals are appropriate? *BMC Medical Research Methodology, 16*, 93. doi:10.1186/s1287401602009

Examples

```
relevance <- matrix(
```

```
  c(4,4,4,3, 4,4,3,4, 3,4,4,4, 4,3,4,4),
  nrow = 4,
  dimnames = list(NULL, paste0("Item", 1:4))
)
fit <- expert_validity(relevance, mode = "relevance", lo = 1, hi = 4, seed = 1)
fit
summary(fit)
```

`gtheory_content` *Generalizability analysis of content-validity ratings*

Description

Decomposes judge ratings of items into item, judge, and residual variance components, then reports how dependably the panel's ratings generalize over judges.

This follows the generalizability-theory treatment of content-validity ratings in Crocker, Llabre, and Miller (1988). Items are the objects of measurement and judges are the facet of generalization, so the question the analysis answers is: *if a different panel of judges of the same size had rated these items, how similar would the conclusions be?*

Two coefficients are reported because they answer different questions:

- The **generalizability coefficient** (relative, `g_coefficient`) concerns the *rank ordering* of items by rated relevance. Use it when the decision is comparative, such as selecting the strongest items from a pool.
- The **dependability coefficient** (absolute, `phi_coefficient`) concerns the *absolute level* of the ratings and is penalized by judge severity differences. Use it when the decision is criterion-referenced, such as whether items clear a fixed relevance standard. Most content-validity decisions are criterion-referenced, so `phi_coefficient` is usually the more relevant of the two.

Because a single rating per judge-item cell cannot separate the judge-by-item interaction from measurement error, the two are reported together as a single residual component. This is a property of the design, not of the estimator.

Usage

```
gtheory_content(
  ratings,
  na.rm = FALSE,
  targets = c(0.7, 0.8, 0.9),
  max_judges = 30
)
```

Arguments

`ratings`	A judges-by-items numeric matrix or data frame: one row per judge, one column per item.
`na.rm`	If `TRUE`, judges with any missing rating are dropped so that a complete crossed design remains, and the number dropped is reported. If `FALSE` (default), missing values are an error.
`targets`	Coefficient targets used for the decision study. Each must lie strictly between 0 and 1.
`max_judges`	Largest panel size shown in the decision-study projection.

Value

An object of class `contentvalid_gtheory`, a list containing:

variance_components Source, degrees of freedom, mean squares, estimated variance component, and percentage of total variance.

coefficients Observed-design generalizability and dependability coefficients with their error variances.

dstudy Projected coefficients across panel sizes.

judges_needed Judges required to reach each target coefficient, for relative and absolute decisions. `NA` means the target is not reachable with any realistic panel, which happens when items are barely distinguished from one another.

settings, design Analysis settings and realized design metadata.

Negative variance estimates

ANOVA estimation can yield negative variance components when a true component is near zero. Negative estimates are truncated to zero for the coefficient calculations, following standard practice, and the untruncated estimate is retained in the `variance_raw` column so the truncation is visible rather than silent.

References

Crocker, L., Llabre, M., & Miller, M. D. (1988). The generalizability of content validity ratings. *Journal of Educational Measurement, 25*(4), 287-299. doi:10.1111/j.17453984.1988.tb00309.x

Brennan, R. L. (2001). *Generalizability Theory.* Springer.

Examples

```
# Six items rated for relevance by eight judges on a 1-4 scale. Items 1-4 are
# clearly relevant, items 5-6 are marginal, and judge 8 is notably severe.
ratings <- rbind(
  c(4, 4, 4, 3, 2, 2), c(4, 4, 3, 4, 2, 1), c(4, 3, 4, 4, 1, 2),
  c(3, 4, 4, 4, 2, 2), c(4, 4, 4, 4, 2, 1), c(4, 3, 4, 3, 1, 2),
  c(4, 4, 3, 4, 2, 2), c(3, 3, 3, 2, 1, 1)
)
dimnames(ratings) <- list(paste0("Judge", 1:8), paste0("Item", 1:6))
gtheory_content(ratings)
```

htc *Hinkin-Tracey correspondence (HTC)*

Description

Computes the Hinkin-Tracey correspondence index for each item. Following Colquitt et al. (2019), HTC is the average definitional-correspondence rating for the intended construct divided by a, the number of rating anchors. Ratings are internally shifted to a 1-to-a metric when a scale such as 0-to-4 is supplied, preserving the meaning of the published formula.

HTC describes definitional correspondence. Higher values indicate that judges see the item as more representative of its intended construct.

Usage

```
htc(
  ratings,
  item_col = "item",
  rater_col = "rater",
  construct_col = "construct",
  rating_col = "rating",
  target_map = NULL,
  target_col = "target_construct",
  scale_min = 1,
  scale_max = 5
)
```

Arguments

ratings	A long-format data.frame containing item, rater, construct, and rating columns.
item_col, rater_col, construct_col, rating_col	Column names.
target_map	Optional named character vector/list mapping item to target.
target_col	Target column used when target_map is NULL.
scale_min, scale_max	Endpoints of the equally spaced integer rating scale (for example, 1 and 5).

Value

A data.frame with item-level target means, usable target-rating counts, and HTC.

References

Hinkin, T. R., & Tracey, J. B. (1999). An analysis of variance approach to content validation. *Organizational Research Methods, 2*(2), 175-186. doi:10.1177/109442819922004

Colquitt, J. A., Sabey, T. B., Rodell, J. B., & Hill, E. T. (2019). Content validation guidelines: Evaluation criteria for definitional correspondence and definitional distinctiveness. *Journal of Applied Psychology, 104*(10), 1243-1265. doi:10.1037/apl0000406

Examples

```
d <- expand.grid(item = "I1", rater = 1:4, construct = c("A", "B"))
d$rating <- c(5, 4, 5, 4, 2, 2, 1, 2)
htc(d, target_map = c(I1 = "A"), scale_min = 1, scale_max = 5)
```

`htd` *Hinkin-Tracey distinctiveness (HTD)*

Description

Computes the Hinkin-Tracey distinctiveness index for each item in a fully crossed, within-judge rating design. For every complete judge, the intended construct rating is contrasted with each orbiting-construct rating. The average of those difference scores is divided by `a - 1`, where `a` is the number of rating anchors. HTD ranges from -1 to 1.

Usage

```
htd(
  ratings,
  item_col = "item",
  rater_col = "rater",
  construct_col = "construct",
  rating_col = "rating",
  target_map = NULL,
  target_col = "target_construct",
  scale_min = 1,
  scale_max = 5
)
```

Arguments

`ratings`	A long-format data.frame containing item, rater, construct, and rating columns.
`item_col`, `rater_col`, `construct_col`, `rating_col`	Column names.
`target_map`	Optional named item-to-target mapping.
`target_col`	Target column used when `target_map` is `NULL`.
`scale_min`, `scale_max`	Endpoints of the equally spaced integer rating scale.

Value

A data.frame containing item-level HTD, the strongest orbiting construct, complete-judge count, and number of target-orbiting pairs.

References

Colquitt, J. A., Sabey, T. B., Rodell, J. B., & Hill, E. T. (2019). *Journal of Applied Psychology, 104*(10), 1243-1265. doi:10.1037/apl0000406

Examples

```
d <- expand.grid(item = "I1", rater = 1:4, construct = c("A", "B", "C"))
d$rating <- c(5,4,5,4, 2,2,1,2, 3,2,2,1)
htd(d, target_map = c(I1 = "A"), scale_min = 1, scale_max = 5)
```

---

interpret_colquitt *Interpret a statistic using Colquitt et al. (2019) norms*

Description

Classifies one or more Psa, Csv, HTC, or HTD values using the empirical percentile bands from Colquitt et al. (2019). These norms were derived from scale-level averages and from naive judges representative of substantive study populations. They should therefore be treated as contextual norms, not pass/fail rules.

When `judge_type = "expert"`, the Colquitt classification is deliberately not applied because the authors caution against using their norms for expert judges.

Usage

```
interpret_colquitt(
  value,
  statistic = c("psa", "csv", "htc", "htd"),
  orbiting_r = NULL,
  judge_type = c("naive", "expert")
)
```

Arguments

`value`	Numeric value(s) to interpret.
`statistic`	One of `"psa"`, `"csv"`, `"htc"`, or `"htd"`.
`orbiting_r`	Optional scalar, or a vector matching `value`, containing the average focal-orbiting correlation. `NULL` uses the overall norms.
`judge_type`	Either `"naive"` or `"expert"`.

Value

A data.frame with the value, benchmark set, interpretation, and an applicability flag.

References

Colquitt, J. A., Sabey, T. B., Rodell, J. B., & Hill, E. T. (2019). *Journal of Applied Psychology, 104*(10), 1243-1265. doi:10.1037/apl0000406

Examples

```
interpret_colquitt(.84, "psa")
interpret_colquitt(.70, "csv", orbiting_r = .40)
```

`ioc` *Item-Objective Congruence (IOC)*

Description

Computes item-objective congruence from expert ratings coded `-1`, `0`, and `+1`. Duplicate item-judge-objective ratings are rejected. Missing ratings may be removed cellwise with transparent effective judge counts.

Usage

```
ioc(ratings, na.rm = FALSE)
```

Arguments

`ratings`	Data frame with columns `item`, `judge`, `objective`, `score`.
`na.rm`	Logical. If `FALSE`, missing scores are an error; if `TRUE`, missing scores are removed within item-objective cells.

Value

A data.frame with item, objective, total rows, effective judge count, missing count, and IOC.

References

Rovinelli, R. J., & Hambleton, R. K. (1977). On the use of content specialists in the assessment of criterion-referenced test item validity. *Dutch Journal of Educational Research, 2*, 49-60.

Turner, R. C., & Carlson, L. (2003). Indexes of item-objective congruence for multidimensional items. *International Journal of Testing, 3*(2), 163-171. doi:10.1207/S15327574IJT0302_5

Examples

```
df <- data.frame(
  item = rep("I1", 6),
  judge = rep(1:3, 2),
  objective = rep(c("A", "B"), each = 3),
  score = c(1,1,1, 0,-1,0)
)
ioc(df)
```

`judge_validity` *Analyze judge and rater heterogeneity in content-validity ratings*

Description

Examines whether content-validity conclusions depend on the particular judges who happened to serve on the panel, rather than reporting only aggregate indices that average heterogeneity away.

The workflow reports four complementary kinds of evidence:

- **Generalizability.** How dependably the panel's ratings would reproduce with a different panel of the same size, via `gtheory_content()`.
- **Severity.** How harsh or lenient each judge is relative to the panel, both in raw rating units and, where estimable, on a logit scale from a many-facet Rasch model fitted as a logistic regression.
- **Response style.** How much each judge differentiates among items, and how much they concentrate on middle or extreme categories.
- **Influence.** Which items would change their CVI-based review status if any single judge were removed from the panel.

A judge flagged for `Review` is not a judge to discard. Disagreement may be substantive expertise rather than error, and removing inconvenient judges is not a validity procedure. The flag identifies where a conclusion rests on one person's ratings and therefore deserves a closer look.

Usage

```
judge_validity(
  ratings,
  lo = 1,
  hi = 4,
  relevance_cut = NULL,
  na.rm = FALSE,
  bias_correct = TRUE,
  severity_cut = 1,
  severity_raw_cut = NULL,
  fit_range = c(0.5, 1.5)
)
```

Arguments

`ratings`	A judges-by-items numeric matrix or data frame of relevance ratings: one row per judge, one column per item.
`lo, hi`	Rating-scale bounds.
`relevance_cut`	Lowest rating treated as relevant. Defaults to `hi - 1`.
`na.rm`	Permit missing ratings. Generalizability analysis additionally requires complete cases and drops incomplete judges, reporting how many.
`bias_correct`	Apply the Wright-Douglas joint-maximum-likelihood bias correction to logit severity estimates. See the estimation note below.

`severity_cut` Absolute logit severity beyond which a judge is flagged for review.

`severity_raw_cut` Absolute severity in rating points beyond which a judge is flagged when logit severity is not estimable. Defaults to a quarter of the scale range. Severity is signed so that positive values mean the judge rates lower than the panel.

`fit_range` Length-2 vector giving the acceptable infit/outfit mean square range. Values outside it flag erratic or overly predictable judges.

Value

An object of class `contentvalid_judge` and `contentvalid_workflow`. Unlike the item-oriented workflows, `results` has **one row per judge**. `scale_summary` describes the panel, and `details` contains the generalizability analysis, the facets model, raw rater effects, and the item-level influence table.

Estimation note

Logit severity comes from a many-facet Rasch model fitted by joint maximum likelihood as a logistic regression, the generalized linear model formulation described by de Boeck and Wilson (2004). Joint maximum likelihood is known to over-disperse facet estimates in small designs. The standard Wright-Douglas `(L-1)/L` correction is applied by default and reported in `settings$bias_correction`, but it reduces rather than removes that bias. Where precise severity calibration matters, marginal maximum likelihood estimation is preferable, and the raw rating-unit severity in `severity_raw` is free of this particular issue.

Severity is estimated from the dichotomized relevance decision, consistent with how the package computes CVI. Judges and items showing no variation in that decision carry no information about relative severity and are excluded from the model, which is reported rather than silent.

References

Crocker, L., Llabre, M., & Miller, M. D. (1988). The generalizability of content validity ratings. *Journal of Educational Measurement, 25*(4), 287-299. doi:10.1111/j.17453984.1988.tb00309.x

Engelhard, G. (1994). Examining rater errors in the assessment of written composition with a many-faceted Rasch model. *Journal of Educational Measurement, 31*(2), 93-112. doi:10.1111/j.17453984.1994.tb00436.x

Linacre, J. M. (1989). *Many-Facet Rasch Measurement.* MESA Press.

de Boeck, P., & Wilson, M. (2004). *Explanatory Item Response Models: A Generalized Linear and Nonlinear Approach.* Springer.

See Also

`gtheory_content()` for the generalizability analysis alone, `expert_validity()` for the item-level expert-panel workflow.

Examples

```
ratings <- rbind(
  c(4, 4, 4, 3, 2, 2), c(4, 4, 3, 4, 2, 1), c(4, 3, 4, 4, 1, 2),
  c(3, 4, 4, 4, 2, 2), c(4, 4, 4, 4, 2, 1), c(4, 3, 4, 3, 1, 2),
  c(4, 4, 3, 4, 2, 2), c(2, 2, 2, 2, 1, 1)
)
dimnames(ratings) <- list(paste0("Judge", 1:8), paste0("Item", 1:6))
fit <- judge_validity(ratings, lo = 1, hi = 4)
fit
summary(fit)
```

---

panel_agreement *Panel-level agreement among expert raters*

Description

Summarizes how consistently a panel rated the whole item set, as one coefficient with a bootstrap interval. This is panel-level evidence. It complements, and does not replace, item-level indices such as I-CVI and modified kappa, which describe one item at a time.

Two coefficients are available:

- `"krippendorff"` (default): Krippendorff's alpha, computed from the coincidence matrix as described by Krippendorff (2011). It works with any number of raters and with missing ratings. Zapf et al. (2016) recommend it specifically when data are ordinal or ratings are missing, which is typical of expert panels. It is a general reliability coefficient (Hayes & Krippendorff, 2007); no publication applying it specifically to content-validity panels was found.
- `"ac1"`: Gwet's (2008) AC1, designed for high-agreement data where kappa-type coefficients fall. It is never the default: Vach and Gerke (2023) show that it rises as ratings concentrate in one category even at a fixed level of agreement, and that it can be non-zero when raters are independent. Its printed output always repeats that critique. AC1 treats the supplied values as unordered categories.

Usage

```
panel_agreement(
  ratings,
  method = c("krippendorff", "ac1"),
  level = c("ordinal", "nominal", "interval"),
  B = 1000,
  alpha = 0.05,
  seed = NULL
)
```

Arguments

`ratings`	A numeric matrix or data frame with raters in rows and items in columns. Missing ratings are allowed.
`method`	`"krippendorff"` (default) or `"ac1"`.
`level`	Measurement level for Krippendorff's alpha: `"ordinal"` (default), `"nominal"`, or `"interval"`. Ignored for AC1.
`B`	Number of bootstrap resamples. Use `0` to skip the interval.
`alpha`	Two-sided error rate for the bootstrap interval; `0.05` gives a 95% interval. This is not Krippendorff's alpha.
`seed`	Optional seed for a reproducible interval.

Value

An object of class `contentvalid_agreement`: a list with `method`, `level`, `estimate`, `ci_low`, `ci_high`, `alpha`, `B`, `n_boot_usable`, `n_items` (items rated by at least two raters), `n_raters`, `percent_agreement` (share of within-item rating pairs that are identical), `interpretation`, and `critique`.

Why a close-agreeing panel can have a low alpha

Alpha compares observed disagreement with the disagreement expected if the same ratings were assigned to items at random. When ratings cluster on a few values, as they do when nearly every item is rated relevant, very little disagreement is expected by chance, so even a few disagreements pull alpha down. The output reports the share of identical rating pairs alongside the coefficient so this pattern is visible rather than misread as a poor panel.

Interval

The interval is a percentile bootstrap that resamples items with all of their ratings intact, the procedure Zapf et al. (2016) evaluated. They found Krippendorff's original bootstrap reached only about 60% coverage because it ignores dependence between raters. Zapf et al. evaluated the procedure for Fleiss' kappa and Krippendorff's alpha; applying it to AC1 is this package's extension. Intervals vary slightly between runs unless `seed` is set.

References

Hayes, A. F., & Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures, 1*(1), 77-89. doi:10.1080/19312450709336664

Krippendorff, K. (2011). *Computing Krippendorff's alpha-reliability.* Annenberg School for Communication, University of Pennsylvania.

Zapf, A., Castell, S., Morawietz, L., & Karch, A. (2016). Measuring inter-rater reliability for nominal data: Which coefficients and confidence intervals are appropriate? *BMC Medical Research Methodology, 16*, 93. doi:10.1186/s1287401602009

Gwet, K. L. (2008). Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology, 61*(1), 29-48. doi:10.1348/000711006X126600

Wongpakaran, N., Wongpakaran, T., Wedding, D., & Gwet, K. L. (2013). A comparison of Cohen's kappa and Gwet's AC1 when calculating inter-rater reliability coefficients: A study conducted with personality disorder samples. *BMC Medical Research Methodology, 13*, 61. doi:10.1186/1471-22881361

Vach, W., & Gerke, O. (2023). Gwet's AC1 is not a substitute for Cohen's kappa: A comparison of basic properties. *MethodsX, 10*, 102212.

Feinstein, A. R., & Cicchetti, D. V. (1990). High agreement but low kappa: I. The problems of two paradoxes. *Journal of Clinical Epidemiology, 43*, 543-549.

See Also

`expert_validity()` for item-level expert-panel evidence.

Examples

```
ratings <- rbind(
  c(4, 4, 3, 2, 4), c(4, 3, 3, 2, 4), c(3, 4, 4, 1, 4), c(4, 4, 3, 2, 3)
)
panel_agreement(ratings, seed = 1)
panel_agreement(ratings, level = "interval", B = 0)
panel_agreement(ratings >= 3, method = "ac1", B = 0)
```

---

plot.contentvalid_expert

Plot expert-panel content-validity results

Description

Draws a mode-specific evidence plot. Relevance mode shows Aiken's V with its score confidence interval and overlays I-CVI as a separate marker. Essentiality mode shows each observed CVR against its item-specific critical CVR. Congruence mode uses a target-versus-strongest-competitor gap plot when a target mapping is available.

Usage

```
## S3 method for class 'contentvalid_expert'
plot(x, show_legend = TRUE, ...)
```

Arguments

`x`	A `contentvalid_expert` object.
`show_legend`	Logical; draw the compact plot key. Default `TRUE`.
`...`	Additional graphical arguments passed to `graphics::plot()`.

Value

The input object invisibly.

Examples

```
relevance <- matrix(
  c(4,4,4,3, 4,4,3,4, 3,4,4,4, 4,3,4,4),
  nrow = 4,
  dimnames = list(NULL, paste0("Item", 1:4))
)
plot(expert_validity(relevance, mode = "relevance", lo = 1, hi = 4,
                     agreement = "none"))
plot(expert_validity(c(10, 8, 6), mode = "essentiality", N = 12))
```

---

plot.contentvalid_expert_power

Plot an expert-panel planning curve

Description

Plot an expert-panel planning curve

Usage

```
## S3 method for class 'contentvalid_expert_power'
plot(x, show_legend = TRUE, ...)
```

Arguments

x	A `contentvalid_expert_power` object.
show_legend	Draw the key identifying each assumed endorsement probability.
...	Passed to `graphics::plot()`.

Value

`x`, invisibly. Called for the plot.

Examples

```
plot(expert_power(n_experts = 3:12, prob = c(0.7, 0.85)))
```

`plot.contentvalid_rating` *Plot Hinkin-Tracey rating evidence*

Description

Provides three complementary views of a construct-rating pretest. `"item"` reproduces the original one-index plot, `"map"` places HTC against HTD to show correspondence and distinctiveness jointly, and `"profile"` draws a target-versus- strongest-competitor gap plot on the original response scale. The latter is a graphical analogue of the mean-rating tables used in Hinkin and Tracey (1999).

Usage

```
## S3 method for class 'contentvalid_rating'
plot(
  x,
  metric = c("htc", "htd"),
  type = c("item", "map", "profile"),
  label = c("review", "all", "none"),
  show_legend = TRUE,
  ...
)
```

Arguments

`x`	A `contentvalid_rating` object.
`metric`	Either `"htc"` or `"htd"` for `type = "item"`.
`type`	One of `"item"`, `"map"`, or `"profile"`.
`label`	Which item labels to draw on the map: `"review"` (default), `"all"`, or `"none"`.
`show_legend`	Logical; draw the compact plot key. Default `TRUE`.
`...`	Additional graphical arguments passed to `graphics::plot()`.

Value

The input object invisibly.

Examples

```
set.seed(12)
d <- expand.grid(item = c("A1", "A2", "B1"), rater = 1:20,
                 construct = c("A", "B", "C"))
d$target_construct <- ifelse(d$item == "B1", "B", "A")
d$rating <- ifelse(d$construct == d$target_construct,
                   pmin(5, pmax(1, round(rnorm(nrow(d), 4.5, .6)))),
                   pmin(5, pmax(1, round(rnorm(nrow(d), 2.0, .7)))))
fit <- rating_validity(d, scale_min = 1, scale_max = 5)
```

```
plot(fit)
plot(fit, type = "map")
plot(fit, type = "profile")
```

`plot.contentvalid_sort` *Plot item-sort evidence*

Description

Draws either the original one-index item plot or a correspondence-distinctiveness evidence map. The map places Psa on the x-axis and Csv on the y-axis so that intended-construct correspondence and distinctiveness can be inspected together. Target-scale means are added as diamonds when available. Colquitt benchmark bands are deliberately not drawn across item points because those norms were developed for scale-level averages rather than individual items.

Usage

```
## S3 method for class 'contentvalid_sort'
plot(
  x,
  metric = c("psa", "csv"),
  type = c("item", "map"),
  label = c("review", "all", "none"),
  show_legend = TRUE,
  ...
)
```

Arguments

`x`	A `contentvalid_sort` object.
`metric`	Either `"psa"` or `"csv"` for `type = "item"`.
`type`	Either `"item"` for the original one-index plot or `"map"` for the correspondence-distinctiveness evidence map.
`label`	Which item labels to draw on the map: `"review"` (default), `"all"`, or `"none"`.
`show_legend`	Logical; draw the compact plot key. Default `TRUE`.
`...`	Additional graphical arguments passed to `graphics::plot()`.

Value

The input object invisibly.

Examples

```
sort_dat <- data.frame(
  item = rep(c("A1", "A2", "A3"), each = 20),
  rater = rep(1:20, 3),
  target_construct = "A",
  assigned_construct = c(
    rep("A", 18), rep("B", 2),
    rep("A", 16), rep("B", 4),
    rep("A", 12), rep("B", 8)
  )
)
fit <- sort_validity(sort_dat)
plot(fit)
plot(fit, type = "map")
```

`plot.contentvalid_sort_power` *Plot exact item-sort planning evidence*

Description

Visualizes either exact Howard-Melloy retention power across planned judge sample sizes or the minimum observed Psa implied by the exact critical target count. The critical view is drawn as a step function over every integer judge count in the displayed range, reflecting the discrete exact-binomial rule. Multiple assumed true target-assignment probabilities are distinguished by line type and plotting symbol rather than color.

Usage

```
## S3 method for class 'contentvalid_sort_power'
plot(
  x,
  type = c("power", "critical"),
  reference_power = NULL,
  show_legend = TRUE,
  ...
)
```

Arguments

`x`	A `contentvalid_sort_power` object.
`type`	Either `"power"` or `"critical"`.
`reference_power`	Optional horizontal reference value for `type = "power"`. No conventional target is imposed by default.
`show_legend`	Logical; draw the compact power-series key. Default `TRUE`.
`...`	Additional graphical arguments passed to `graphics::plot()`.

Value

The input object invisibly.

Examples

```
plan <- sort_power(N = c(20, 30, 40), true_p = c(.60, .70, .80))
plot(plan)
plot(plan, type = "critical")
```

`plot.contentvalid_structure` *Plot an expert content map*

Description

Plots the multidimensional scaling content map from `content_structure()`, with each item positioned by expert-perceived similarity and labeled by its blueprint cell. Items that sit away from others sharing their cell are the ones experts did not group as the blueprint expects.

Usage

```
## S3 method for class 'contentvalid_structure'
plot(x, show_legend = TRUE, ...)
```

Arguments

`x`	A `contentvalid_structure` object.
`show_legend`	Draw the blueprint-cell key.
`...`	Passed to `graphics::plot()`.

Value

`x`, invisibly. Called for the plot.

Examples

```
items <- paste0("I", 1:6)
blueprint <- c(rep("Autonomy", 3), rep("Competence", 3))
sim <- matrix(1, 6, 6, dimnames = list(items, items))
sim[1:3, 1:3] <- 5
sim[4:6, 4:6] <- 5
diag(sim) <- 5
plot(content_structure(sim, membership = blueprint))
```

qfactor_content *Q-factor helper for content adequacy (comparator)*

Description

Builds an item-by-item Q-correlation matrix from rating data and runs a factor extraction (PCA by default), following the content-adequacy approach of Schriesheim et al. (1993): judges rate every item against every construct definition, and items that measure the same construct correlate across those ratings. Schriesheim et al. (1999) compared this approach empirically with other content-adequacy methods and found substantial similarity along with some differences. This comparator is retained for compatibility and exploratory use; it is not part of the recommended sort, rating, or expert-panel workflows.

Usage

```
qfactor_content(
  ratings,
  item_col = "item",
  rater_col = "rater",
  construct_col = "construct",
  rating_col = "rating",
  k_factors = NULL,
  method = c("pca", "pa"),
  retention = c("parallel", "kaiser"),
  parallel_criterion = c("mean", "percentile"),
  percentile = 95,
  n_iter = 100,
  seed = NULL
)
```

Arguments

ratings	A data.frame with columns for item, rater, construct, rating.
item_col	Name of the item column. Default "item".
rater_col	Name of the rater column. Default "rater".
construct_col	Name of the construct column. Default "construct".
rating_col	Name of the rating column. Default "rating".
k_factors	Optional integer: number of factors to extract. When supplied, retention is ignored.
method	"pca" (default) or "pa" (principal axis; uses SMCs as initial communalities).
retention	How to choose the number of factors when k_factors is NULL: "parallel" (default) or "kaiser". See the section below.
parallel_criterion	What parallel analysis compares against: "mean" (default, Horn) or "percentile" (Glorfeld).

percentile	Upper percentile used when parallel_criterion = "percentile". Default 95, as in Glorfeld (1995).
n_iter	Number of random data sets for parallel analysis.
seed	Optional seed that makes parallel analysis reproducible.

Value

A list with components:

- cor_Q: item-by-item correlation matrix,
- eigen: eigenvalues of cor_Q,
- k: number of factors used,
- loadings: matrix of factor loadings,
- method: the extraction method,
- retention: "parallel", "kaiser", or "fixed" when k_factors was supplied,
- k_suggested: the number of factors the retention rule suggested, which can be 0,
- parallel_eigen: the comparison eigenvalues parallel analysis used, or NULL when parallel analysis was not run,
- parallel_criterion: "mean" or "percentile", or NA when parallel analysis was not run,
- percentile: the percentile used, or NA for the mean criterion.

Number of factors

Unless k_factors is supplied, retention sets the number of factors:

- "parallel" (default): Horn's (1965) parallel analysis. The eigenvalues of the Q-correlation matrix are compared, in order, with eigenvalues from random normal data of the same size and with the same missing cells. Factors are retained while the observed eigenvalue is larger. Zwick and Velicer (1986) found parallel analysis among the most accurate rules. Results vary slightly between runs unless seed is set.
- "kaiser": retain eigenvalues greater than 1. This was the default before contentvalidR 0.3.0 and remains available so earlier results can be reproduced. It is never the default: Zwick and Velicer (1986) found that it severely overestimates the number of components, and choosing it prints a message saying so.

parallel_criterion chooses what the observed eigenvalues are compared against:

- "mean" (default): the mean simulated eigenvalue, as in Horn (1965). This is the rule Zwick and Velicer (1986) evaluated and the one contentvalidR 0.3.0 shipped.
- "percentile": the upper percentile of the simulated eigenvalue distribution, following Glorfeld (1995). Glorfeld noted that Horn's procedure, while relatively accurate, still tends to indicate the retention of one or two more factors than is warranted, and proposed comparing against a chosen upper percentile instead. It is the stricter rule and retains no more factors than the mean criterion on the same simulation.

Both rules use the eigenvalues of the full Q-correlation matrix, with 1s on the diagonal, whichever extraction method is used. At least one factor is always extracted; k_suggested shows when a rule suggested none.

References

Glorfeld, L. W. (1995). An improvement on Horn's parallel analysis methodology for selecting the correct number of factors to retain. *Educational and Psychological Measurement, 55*(3), 377-393. doi:10.1177/0013164495055003002

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika, 30*(2), 179-185. doi:10.1007/BF02289447

Schriesheim, C. A., Powers, K. J., Scandura, T. A., Gardiner, C. C., & Lankau, M. J. (1993). Improving construct measurement in management research: Comments and a quantitative approach for assessing the theoretical content adequacy of paper-and-pencil survey-type instruments. *Journal of Management, 19*(2), 385-417. doi:10.1177/014920639301900208

Schriesheim, C. A., Cogliser, C. C., Scandura, T. A., Lankau, M. J., & Powers, K. J. (1999). An empirical comparison of approaches for quantitatively assessing the content adequacy of paper-and-pencil measurement instruments. *Organizational Research Methods, 2*(2), 140-156. doi:10.1177/109442819922002

Zwick, W. R., & Velicer, W. F. (1986). Comparison of five rules for determining the number of components to retain. *Psychological Bulletin, 99*(3), 432-442. doi:10.1037/00332909.99.3.432

Examples

```
set.seed(1)
df <- data.frame(
  item = rep(paste0("I",1:6), each = 30),
  rater = rep(1:10, times = 18),
  construct = rep(rep(LETTERS[1:3], each = 10), times = 6),
  rating = rnorm(180)
)
qf <- qfactor_content(df, seed = 1)
qf$k
str(qf$loadings)

# Glorfeld's stricter comparison, on the same simulation.
qfactor_content(df, parallel_criterion = "percentile", seed = 1)$k_suggested
```

`rating_validity` *Analyze a Hinkin-Tracey construct-rating content-validity pretest*

Description

Provides the recommended user-facing workflow for a fully crossed construct-rating study. Judges rate each item against its intended construct definition and one or more orbiting definitions. `rating_validity()` combines:

- Hinkin-Tracey correspondence (HTC),
- Hinkin-Tracey distinctiveness (HTD),
- one-way repeated-measures ANOVA (with Greenhouse-Geisser correction) for each item, and
- planned paired target-versus-orbiting contrasts.

Item-level output is diagnostic rather than a coefficient dump: it identifies the strongest competing construct, describes why an item was flagged, and labels statistical screening decisions `"Retain"`, `"Review"`, or `"Insufficient data"`. `"Review"` is not an instruction to delete the item.

Colquitt et al. (2019) norms are applied only to **target-scale averages** of HTC and HTD, matching the level at which those empirical benchmarks were constructed. The labels are suppressed for expert judges.

Usage

```
rating_validity(
  ratings,
  item_col = "item",
  rater_col = "rater",
  construct_col = "construct",
  rating_col = "rating",
  target_map = NULL,
  target_col = "target_construct",
  scale_min = 1,
  scale_max = 5,
  alpha = 0.05,
  adjust = c("none", "holm"),
  orbiting_r = NULL,
  judge_type = c("naive", "expert")
)
```

Arguments

`ratings`	Long-format rating data.
`item_col`, `rater_col`, `construct_col`, `rating_col`	Column names.
`target_map`	Optional named item-to-target mapping.
`target_col`	Target column used when `target_map` is `NULL`.
`scale_min`, `scale_max`	Endpoints of the equally spaced integer rating scale.
`alpha`	Significance level for item-level inferential screening.
`adjust`	Planned-contrast p-value adjustment: `"none"` (historical planned-comparison logic) or `"holm"`.
`orbiting_r`	Optional average focal-orbiting correlation. For multiple target scales, use a named numeric vector keyed by target.
`judge_type`	Either `"naive"` or `"expert"`. Colquitt normative labels are not applied to expert-judge data.

Value

An object of class `contentvalid_rating` and `contentvalid_workflow`. All flagship workflow objects expose the common components `results`, `scale_summary`, `settings`, `design`, and `details`. Planned contrasts live in `details$contrasts`; the historical top-level `contrasts` component is

retained as a compatibility alias. Item-level `results` include a standardized `status` field while retaining the method-specific `recommendation` field.

References

Hinkin, T. R., & Tracey, J. B. (1999). An analysis of variance approach to content validation. *Organizational Research Methods, 2*(2), 175-186. doi:10.1177/109442819922004

Colquitt, J. A., Sabey, T. B., Rodell, J. B., & Hill, E. T. (2019). Content validation guidelines: Evaluation criteria for definitional correspondence and definitional distinctiveness. *Journal of Applied Psychology, 104*(10), 1243-1265. doi:10.1037/apl0000406

Examples

```
set.seed(12)
d <- expand.grid(item = c("A1", "A2", "B1"), rater = 1:20,
                 construct = c("A", "B", "C"))
d$target_construct <- ifelse(d$item == "B1", "B", "A")
d$rating <- ifelse(d$construct == d$target_construct,
                   pmin(5, pmax(1, round(rnorm(nrow(d), 4.5, .6)))),
                   pmin(5, pmax(1, round(rnorm(nrow(d), 2.0, .7)))))
fit <- rating_validity(d, scale_min = 1, scale_max = 5)
fit
summary(fit)
```

---

reproducibility_phi *Between-pretest reproducibility (phi) of binary decisions*

Description

Auxiliary compatibility diagnostic. Cross-tabulates retention decisions for the same items across two pretests and reports signed phi and Pearson's chi-square test without Yates correction. The full 2 x 2 table is retained even when one response level is absent.

Usage

```
reproducibility_phi(sig1, sig2)
```

Arguments

`sig1`	Logical vector of retention decisions from pretest 1.
`sig2`	Logical vector of retention decisions from pretest 2.

Value

A list containing the 2 x 2 table, signed phi, chi-square, and p-value.

Examples

```
sig1 <- c(TRUE, TRUE, FALSE, FALSE)
sig2 <- c(TRUE, FALSE, FALSE, TRUE)
reproducibility_phi(sig1, sig2)
```

`signal_detection` *Signal-detection summary for binary retention decisions*

Description

Auxiliary compatibility diagnostic. Compares a logical vector of pretest retention decisions with a logical ground-truth criterion (for example, later CFA retention). Reports a correctly oriented confusion matrix, accuracy, sensitivity, specificity, signed phi, and Pearson's chi-square test without Yates correction.

The comparison follows the validation design of Anderson and Gerbing (1991), who checked pretest assessments of items' substantive validity against how those items later performed in a confirmatory factor analysis.

Usage

```
signal_detection(predicted, actual)
```

Arguments

`predicted`	Logical vector of predicted retention decisions.
`actual`	Logical vector of criterion retention decisions.

Value

A list containing the confusion matrix and diagnostic statistics.

References

Anderson, J. C., & Gerbing, D. W. (1991). Predicting the performance of measures in a confirmatory factor analysis with a pretest assessment of their substantive validities. *Journal of Applied Psychology, 76*(5), 732-740. doi:10.1037/00219010.76.5.732

Examples

```
predicted <- c(TRUE, TRUE, FALSE, FALSE)
actual    <- c(TRUE, FALSE, TRUE, FALSE)
signal_detection(predicted, actual)
```

`similarity_from_sort` *Derive item similarities from an item-sort task*

Description

Builds an item-by-item similarity matrix from item-sort data, where the similarity of two items is the proportion of judges who assigned them to the same construct.

This lets `content_structure()` be used when a study collected a sorting task rather than the pairwise similarity ratings of Sireci and Geisinger (1992).

Usage

```
similarity_from_sort(
  assignments,
  item_col = "item",
  rater_col = "rater",
  assigned_col = "assigned_construct"
)
```

Arguments

`assignments`	A long-format data frame of sort assignments.
`item_col`, `rater_col`, `assigned_col`	Column names.

Value

A square, symmetric item-by-item matrix of co-assignment proportions, with attribute `"n_pairs"` giving the number of judges contributing to each cell.

Weaker evidence than a similarity task

Co-assignment similarity is coarser than a direct similarity rating. A sort forces every item into exactly one construct, so two items placed in different constructs record zero similarity no matter how closely related a judge considers them, and the recovered structure is constrained toward the construct set the sorting task offered. Structure recovered this way is evidence about how judges sorted, which is a weaker basis for claims about perceived content structure than pairwise similarity ratings collected for that purpose.

References

Sireci, S. G., & Geisinger, K. F. (1992). Analyzing test content using cluster analysis and multidimensional scaling. *Applied Psychological Measurement, 16*(1), 17-31. doi:10.1177/014662169201600102

Examples

```
sorts <- data.frame(
  item = rep(paste0("I", 1:4), each = 5),
  rater = rep(1:5, times = 4),
  assigned_construct = c(rep("A", 5), rep("A", 5), rep("B", 5), rep("B", 5))
)
similarity_from_sort(sorts)
```

---

simulate_anova_power *Legacy independent-groups ANOVA power simulator*

Description

Simulates a balanced **independent-groups** one-way ANOVA. This helper is retained for backward compatibility but does not represent the standard within-judge Hinkin-Tracey design used by `rating_validity()`. It is an auxiliary compatibility helper and is not a release-defining workflow.

Usage

```
simulate_anova_power(
  n_raters = 30,
  mean_diff = 0.6,
  sd = 1,
  k_constructs = 5,
  reps = 1000,
  alpha = 0.05
)
```

Arguments

`n_raters`	Number of raters per construct (balanced).
`mean_diff`	Target mean minus other-construct means.
`sd`	Within-cell standard deviation.
`k_constructs`	Number of constructs.
`reps`	Number of simulation replications.
`alpha`	Significance level.

Value

Estimated power (numeric in [0, 1]).

Examples

```
simulate_anova_power(n_raters = 20, mean_diff = 0.5, sd = 1, k_constructs = 4, reps = 100)
```

simulate_csv_power *Legacy simulation of item-sort target-count power*

Description

Auxiliary compatibility helper. For supported exact planning, prefer `sort_power()`, which does not require Monte Carlo simulation.

Usage

```
simulate_csv_power(N = 20, true_p = 0.65, reps = 2000, alpha = 0.05)
```

Arguments

N	Number of judges per item.
true_p	True assignment probability to the target construct.
reps	Number of simulation replications.
alpha	Significance level.

Value

Estimated power (a number between 0 and 1).

Examples

```
simulate_csv_power(N = 20, true_p = 0.65, reps = 100, alpha = 0.05)
```

sort_power *Exact power for the item-sort target-count rule*

Description

Computes the exact probability that an item will meet the Howard-Melloy target-count criterion for a planned judge sample size and an assumed true target-assignment probability. This is a binomial calculation, not a simulation.

Usage

```
sort_power(N, true_p, p0 = 0.5, alpha = 0.05)
```

Arguments

N	Positive integer judge sample size(s).
true_p	Assumed true probability that a judge assigns the item to its intended construct. May be scalar or vector.
p0	Null target-assignment probability. Default `0.5`.
alpha	Significance level. Default `0.05`.

Value

An object of class `contentvalid_sort_power` containing an exact planning table.

Examples

```
sort_power(N = c(20, 30, 40), true_p = .70)
sort_power(N = 30, true_p = c(.60, .70, .80))
```

---

sort_validity *Analyze an item-sort content-validity pretest*

Description

Provides the recommended user-facing workflow for item-sort studies. At the item level, `sort_validity()` combines Anderson and Gerbing's (1991) Psa and Csv statistics with the exact target-count significance test recommended by Howard and Melloy (2016). Items meeting the exact criterion are labeled `"Retain"`; items that do not meet it are labeled `"Review"`, not automatically `"Delete"`.

At the target-scale level, Psa and Csv are averaged across items and interpreted using the empirical percentile norms from Colquitt et al. (2019). This mirrors how those norms were constructed. The Colquitt categories are descriptive benchmarks rather than pass/fail rules.

Usage

```
sort_validity(
  assignments,
  item_col = "item",
  rater_col = "rater",
  assigned_col = "assigned_construct",
  target_col = "target_construct",
  p0 = 0.5,
  alpha = 0.05,
  orbiting_r = NULL,
  judge_type = c("naive", "expert"),
  proportion_ci = c("wilson", "agresti_coull", "exact", "none")
)
```

Arguments

`assignments`	A data.frame containing item-sort responses.
`item_col`, `rater_col`, `assigned_col`, `target_col`	Column names for the item, rater, assigned construct, and intended target construct.
`p0`	Null target-assignment probability for the exact binomial test. Default `0.5`, following Howard and Melloy (2016).
`alpha`	Significance level. Default `0.05`.

orbiting_r	Optional average correlation between each focal/target scale and its orbiting scales. For one target, supply one correlation. For multiple targets, supply a named numeric vector keyed by target construct. If omitted, the overall Colquitt et al. norms are used.
judge_type	Either "naive" (the Anderson-Gerbing/Colquitt design) or "expert". Colquitt benchmark labels are not applied to expert judges.
proportion_ci	Interval method for Psa: "wilson" (default), "agresti_coull", "exact", or "none". The interval uses the same alpha as the exact test. See ci in cvi() for the methods and the evidence for each.

Value

An object of class contentvalid_sort and contentvalid_workflow. All flagship workflow objects expose the common components results, scale_summary, settings, design, and details. Item-level results include a standardized status field while retaining the method-specific recommendation field. print(), summary(), and plot() provide user-facing interpretation.

References

Anderson, J. C., & Gerbing, D. W. (1991). Predicting the performance of measures in a confirmatory factor analysis with a pretest assessment of their substantive validities. *Journal of Applied Psychology, 76*(5), 732-740. doi:10.1037/00219010.76.5.732

Howard, M. C., & Melloy, R. C. (2016). Evaluating item-sort task methods: The presentation of a new statistical significance formula and methodological best practices. *Journal of Business and Psychology, 31*(1), 173-186. doi:10.1007/s108690159404y

Colquitt, J. A., Sabey, T. B., Rodell, J. B., & Hill, E. T. (2019). Content validation guidelines: Evaluation criteria for definitional correspondence and definitional distinctiveness. *Journal of Applied Psychology, 104*(10), 1243-1265. doi:10.1037/apl0000406

Examples

```
sort_dat <- data.frame(
  item = rep(c("A1", "A2", "A3"), each = 20),
  rater = rep(1:20, 3),
  target_construct = "A",
  assigned_construct = c(
    rep("A", 18), rep("B", 2),
    rep("A", 16), rep("B", 4),
    rep("A", 12), rep("B", 8)
  )
)
fit <- sort_validity(sort_dat)
fit
summary(fit)
```

Index