Package: misskappa (via r-universe)

November 22, 2026

Type Package

Title Agreement Coefficients with Missing Ratings

Version 3.0.0

Description Estimates chance-corrected agreement coefficients (Cohen's, Conger's, and Fleiss' kappa and the Brennan-Prediger coefficient; Conger (1980) <doi:10.1037/0033-2909.88.2.322>) for any number of raters, with nominal, linear, or quadratic weights, when some ratings are missing. Missing ratings are handled by inverse-probability weighting (Horvitz and Thompson (1952) <doi:10.1080/01621459.1952.10483446>) when they are missing completely at random, and by categorical or robust normal-theory full-information maximum likelihood when they are missing at random (Little and Rubin (2019) <doi:10.1002/9781119482260>). Every estimate comes with an influence-function standard error, giving confidence intervals and Wald tests that compare coefficients across groups, raters, or estimators. Also handles counts-format data, multirater g-wise agreement (Moss (2024) <doi:10.1007/s11336-023-09945-2>), and vector-valued ratings.

URL https://jonasmoss.github.io/misskappa/, https://github.com/JonasMoss/misskappa

BugReports https://github.com/JonasMoss/misskappa/issues

License MIT + file LICENSE

Encoding UTF-8

SystemRequirements C++17

LazyData true

Depends R (>= 4.0.0)

Imports Rcpp, stats, utils

LinkingTo Rcpp, RcppEigen

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

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/load source

Config/roxygen2/markdown true

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Jonas Moss [aut, cre] (ORCID:
<https://orcid.org/0000-0002-6876-6964>)

Maintainer Jonas Moss <jonas.moss.statistics@gmail.com>

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

Date/Publication 2026-11-21 19:10:02 UTC

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

RemoteRef HEAD

RemoteSha 6518bebba647e12abe18710a9bc3bd0393bd242c

Contents

agreement 2
agreement_counts 5
agreement_test 7
as.data.frame.misskappa_estimate 8
confint.misskappa_estimate 8
dat.fleiss1971 10
dat.gwet2014 11
dat.klein2018 11
dat.mcduff2019 12
dat.tenhove2025 13
dat.vanbelle2019 14
dat.zapf2016 15
influence.misskappa_estimate 15
ratings_to_counts 16

Index **18**

`agreement` *Weighted agreement coefficients with missing data*

Description

Estimates Conger, Fleiss, and Brennan-Prediger agreement coefficients (with two raters, Conger is Cohen's kappa) from a subjects-by-raters matrix in which some ratings may be missing. Choose the estimator with 'estimator':

- '"ipw"' (default, recommended): inverse-probability weighting. Consistent under MCAR, also with raters observed at different rates; works with any 'weight'.

- ‘"cat_fiml"‘: categorical full-information maximum likelihood. Consistent under MAR; feasible for few raters and categories.
- ‘"pairwise"‘: moment estimator for quadratic weights, MCAR.
- ‘"nt_fiml"‘: robust normal-theory FIML for quadratic weights, suited to MAR with roughly normal scores. Falls back to ‘"pairwise"‘ when the fit fails; see the section on the fallback.
- ‘"nt_fiml_cs"‘: ‘"nt_fiml"‘ under the compound-symmetry covariance $\Sigma = \psi J + eI$; needs only a connected rater co-observation graph, so it handles planned-incomplete designs.

The last three treat ratings as numeric scores and use the quadratic weight.

Usage

```
agreement(
  x,
  estimator = c("ipw", "cat_fiml", "pairwise", "nt_fiml", "nt_fiml_cs"),
  weight = c("nominal", "linear", "quadratic", "hubert"),
  g = 2L,
  values = NULL,
  categories = NULL,
  em_options = list(),
  fallback = NULL,
  ...
)
```

Arguments

x	A subjects-by-raters matrix or data frame of ratings, or a subjects-by-raters-by-features array of vector-valued ratings. ‘NA‘ marks missing entries.
estimator	‘"ipw"‘ (default), ‘"cat_fiml"‘, ‘"pairwise"‘, ‘"nt_fiml"‘, or ‘"nt_fiml_cs"‘.
weight	‘"nominal"‘ (default), ‘"linear"‘, or ‘"quadratic"‘. The scored estimators always use ‘"quadratic"‘. ‘"hubert"‘ is an all-raters-equal kernel for ‘g > 2‘. For vector-valued ratings it selects the component loss (Hamming, absolute, or squared).
g	Number of raters the disagreement kernel compares at once. ‘2‘ (default) is ordinary kappa. Quadratic agreement does not depend on ‘g‘.
values	Optional numeric scores, one per category in scale order. Defaults to the categories themselves when numeric, else ‘1:C‘.
categories	Optional vector declaring the full rating scale in order, for example ‘1:5‘ or ‘c("low", "mid", "high")‘.
em_options	Named list of EM settings for the likelihood estimators: ‘tol‘ and ‘max_iter‘, and for ‘"cat_fiml"‘ also ‘prune_tol‘, ‘start_alpha‘, ‘info_rcond‘, and ‘flatten‘. ‘info_rcond‘ is the relative eigenvalue cutoff of the information matrix used for the standard errors. ‘flatten > 0‘ adds a small Dirichlet pseudo-count that makes the fitted table unique when the likelihood is flat; the estimate barely moves but standard errors become conservative. Settings an estimator does not use are an error.

`fallback` For ‘estimator = "nt_fiml"‘ only: ‘"pairwise"‘ (the default) returns the ‘"pairwise"‘ estimate when the fit fails or does not converge; ‘"none"‘ raises the error. See the section on the fallback.

`...` Arguments for vector-valued ratings: ‘W‘ (a feature metric for the squared loss), ‘ranges‘ (a features-by-2 matrix of score bounds, for Brennan-Prediger), ‘feature_weights‘, and ‘loss = "rms"‘. For ‘g > 2‘: ‘max_chance_tuples‘ (default ‘5e6‘). Anything else is an error.

Value

A ‘misskappa_estimate‘: the estimates, their covariance matrix, the number of subjects ‘n‘, the rating scale ‘categories‘, and the per-subject influence rows ‘psi‘, with ‘vcov = crossprod(psi) / n^2‘. ‘"cat_fiml"‘ fits also carry two diagnostics per coefficient: ‘null_frac‘, the share of the coefficient’s gradient that lies in directions the likelihood does not identify, and ‘se_cutoff_ratio‘, the standard error recomputed with ‘info_rcond‘ lowered 100-fold divided by the reported one (values well above 1 mean the standard error leans on weakly identified directions of a sparse table). Methods: ‘print‘, ‘coef‘, ‘vcov‘, ‘confint‘, ‘as.data.frame‘, and ‘stats::influence‘.

Rating scale

Ratings may be numbers, strings, logicals, or factors. The scale defaults to the shared factor levels, or else the sorted observed ratings. Text ratings have no order of their own, so linear and quadratic weights and the scored estimators need ‘categories‘ (or factors with shared levels) for them. Pass ‘categories‘ to declare the full scale including categories nobody used (‘categories = 1:5‘ on a 1-5 scale where no one chose 5). The scale sets Brennan-Prediger’s chance baseline and the range that scales the linear and quadratic weights. ‘"nt_fiml"‘ and ‘"nt_fiml_cs"‘ report Brennan-Prediger only when ‘categories‘, ‘values‘, or factor levels declare the scale; ‘"pairwise"‘ reports Conger and Fleiss only.

Missing data

Subjects with no ratings are dropped. Every rater must be observed, and every rater pair must rate at least one subject together (every g-tuple for ‘g > 2‘), otherwise the coefficient is not identified and the call errors. ‘"nt_fiml_cs"‘ needs only a connected co-observation graph.

Fallback for ‘"nt_fiml"‘

The normal-theory likelihood breaks down when the fitted covariance of the ratings is singular or nearly so, for example when two raters agree perfectly on the subjects they share. Kappa itself is still well defined there, but the maximum-likelihood fit and its sandwich standard error are not. This is common in small samples with high agreement and in planned designs where each rater pair shares few subjects. On complete data ‘"nt_fiml"‘ equals ‘"pairwise"‘ and never fails.

By default (‘fallback = "pairwise"‘), a failed or non-converged ‘"nt_fiml"‘ fit is replaced by the ‘"pairwise"‘ estimate of the same coefficient, with a warning and the reason recorded in ‘fit$fallback‘. ‘"pairwise"‘ is the moment estimator that ‘"nt_fiml"‘ reduces to on complete data, so the combination is consistent under MCAR everywhere and keeps ‘"nt_fiml"‘’s advantage under MAR whenever the fit succeeds. In the simulation study of Moss (in preparation) the combination had smaller bias at n = 40 than either estimator alone, because ‘"nt_fiml"‘ fails selectively on high-agreement samples. Use ‘fallback = "none"‘ to get the error instead.

Generalizations

‘g > 2‘ uses g-wise kernels that compare ‘g‘ raters at once (with ‘"ipw"‘ or ‘"cat_fiml"‘). A subjects-by-raters-by-features array gives vector-valued agreement; see the beyond-pairwise article.

References

Moss, J. (in preparation). *Agreement coefficients with missing ratings.*

Moss, J. (in preparation). *Quadratically weighted agreement coefficients.*

See Also

[agreement_counts()] for counts-format data, [agreement_test()] to compare coefficients, [confint.misskappa_estimate()].

Examples

```
# Five raters, three categories, some ratings missing.
agreement(dat.klein2018)

# Ordered categories with linear weights, on a declared 1-4 scale.
agreement(dat.klein2018, weight = "linear", categories = 1:4)

# Quadratic agreement on a 1-5 scale under MAR, normal theory.
agreement(dat.zapf2016, estimator = "nt_fiml", categories = 1:5)

# A planned-incomplete design: some rater pairs never meet, so only the
# compound-symmetry estimator is identified.
agreement(dat.tenhove2025, estimator = "nt_fiml_cs")
```

---

agreement_counts *Agreement coefficients for counts-format data*

Description

Estimates Fleiss and Brennan-Prediger coefficients when each row is a subject and each column a category, with ‘x[i, k]‘ the number of raters who put subject ‘i‘ in category ‘k‘. Rows may have different totals, which is how missing ratings show up in this format. Raters are anonymous, so they are treated as exchangeable and Conger is not reported. For rater-identified data use [agreement()]; [ratings_to_counts()] converts it to this format.

- ‘"pairwise"‘ (default, recommended): the moment estimator of Fleiss and Cuzick (1979), extended to linear and quadratic weights. It averages agreement over the rater pairs within each subject, the counts-format analogue of ‘"pairwise"‘ in [agreement()]. Consistent under MCAR, any ‘weight‘.
- ‘"cat_fiml"‘: full-information maximum likelihood over the category compositions, consistent under MAR. It is the ‘"cat_fiml"‘ of [agreement()] with the raters constrained to be exchangeable (the same likelihood), which is why it shares the name.

- ‘"nt_fiml_cs"‘: normal-theory compound-symmetry FIML, quadratic weight only.

Rows with fewer than two ratings carry no agreement information; at least one row needs two or more.

Usage

```
agreement_counts(
  x,
  estimator = c("pairwise", "cat_fiml", "nt_fiml_cs"),
  weight = c("nominal", "linear", "quadratic"),
  values = NULL,
  r_total = NULL,
  em_options = list()
)
```

Arguments

x	A subjects-by-categories matrix of non-negative counts.
estimator	‘"pairwise"‘ (default), ‘"cat_fiml"‘, or ‘"nt_fiml_cs"‘.
weight	‘"nominal"‘ (default), ‘"linear"‘, or ‘"quadratic"‘. ‘"nt_fiml_cs"‘ always uses ‘"quadratic"‘.
values	Optional numeric category scores for the linear and quadratic weights. Defaults to ‘1:C‘.
r_total	For ‘"cat_fiml"‘: the number of raters each subject would have with no missing ratings. Defaults to the largest row total.
em_options	Named list of EM settings for the likelihood estimators: ‘tol‘ and ‘max_iter‘, and for ‘"cat_fiml"‘ also ‘prune_tol‘, ‘start_alpha‘, and ‘info_rcond‘.

Value

A ‘misskappa_estimate‘ with ‘Fleiss‘ and ‘Brennan-Prediger‘; see [agreement()].

References

Fleiss, J. L., & Cuzick, J. (1979). The reliability of dichotomous judgments: Unequal numbers of judges per subject. *Applied Psychological Measurement*, 3(4), 537-542.

Moss, J. (in preparation). *Quadratically weighted agreement coefficients.*

Examples

```
# Fleiss (1971): 30 subjects, 6 raters, 5 diagnostic categories.
agreement_counts(dat.fleiss1971)

# Ordered categories with quadratic weights.
agreement_counts(dat.fleiss1971, weight = "quadratic")
```

`agreement_test` *Test equality of an agreement coefficient across fits*

Description

Wald test that a coefficient is equal across two or more fits, or equal to ‘value‘ for a single fit. With more than two fits it tests that all are equal (‘G - 1‘ degrees of freedom).

Fits are independent samples by default, so their variances add. Use ‘paired = TRUE‘ for fits on the same subjects (two estimators, two attributes, two rater groups, ...): the dependence then comes from the per-subject influence rows, matched by the row names of the rating data, or by row number when there are none.

Usage

```
agreement_test(..., coef = NULL, paired = FALSE, value = 0)
```

Arguments

`...`	Fits from [agreement()] or [agreement_counts()]. Names label the output.
`coef`	Coefficient to compare. Defaults to the first one (‘"Conger"‘ for [agreement()], ‘"Fleiss"‘ for [agreement_counts()]).
`paired`	‘FALSE‘ (default) for independent samples, ‘TRUE‘ for fits on the same subjects.
`value`	Null value for the coefficient (one fit) or the difference (two fits). Default ‘0‘.

Value

An ‘htest‘ object.

Examples

```
# Two studies with different subjects.
agreement_test(gwet  = agreement(dat.gwet2014),
               klein = agreement(dat.klein2018))

# Two rater pairs on the same 50 subjects.
agreement_test(AB = agreement(dat.zapf2016[, 1:2], estimator = "pairwise"),
               CD = agreement(dat.zapf2016[, 3:4], estimator = "pairwise"),
               paired = TRUE)
```

`as.data.frame.misskappa_estimate` *Coerce a misskappa fit to a data frame*

Description

One row per coefficient with the estimator, the estimate, its standard error, and confidence limits from [confint()][confint.misskappa_estimate]. Fits from several calls can be stacked with ‘rbind()‘.

Usage

```
## S3 method for class 'misskappa_estimate'
as.data.frame(x, row.names = NULL, optional = FALSE, level = 0.95, ...)
```

Arguments

`x`	A ‘misskappa_estimate‘ object.
`row.names, optional`	Passed for S3 conformance. ‘row.names‘ sets the row names of the result.
`level`	Confidence level for the ‘lower‘ and ‘upper‘ columns.
`...`	Further arguments to [confint()][confint.misskappa_estimate], such as ‘method‘ or ‘transform‘.

Value

A data frame with columns ‘estimator‘, ‘coefficient‘, ‘estimate‘, ‘se‘, ‘lower‘, and ‘upper‘.

Examples

```
fit <- agreement(dat.gwet2014, estimator = "ipw")
as.data.frame(fit)
```

`confint.misskappa_estimate` *Confidence intervals for misskappa coefficients*

Description

Wald or exact-delete-one jackknife confidence intervals for a ‘misskappa_estimate‘. Wald intervals use the estimator covariance and a normal critical value. Jackknife intervals transform every complete-data leave-one-subject-out estimate directly, use the ordinary delete-one jackknife standard error, and use a Student t critical value with ‘n - 1‘ degrees of freedom.

Besides the natural and Fisher scales, ‘transform = "log_complement"‘ uses the upper-bound-only transform ‘-log(1 - theta)‘, which stays defined for coefficients below -1. At an exact boundary (perfect agreement), a degenerate transformed jackknife is replaced by a one-sided hybrid: a Clopper–Pearson upper limit for the probability of any item disagreement is divided by a delete-one t lower limit for chance disagreement. When that fallback is unavailable the interval is returned as ‘NA‘ with a warning.

Usage

```
## S3 method for class 'misskappa_estimate'
confint(
  object,
  parm = NULL,
  level = 0.95,
  method = c("wald", "jackknife"),
  transform = c("none", "fisher", "log_complement"),
  ...
)
```

Arguments

`object`	A ‘misskappa_estimate‘ object.
`parm`	Optional subset of coefficients (names or indices), defaults to all.
`level`	Confidence level.
`method`	Either ‘"wald"‘ or ‘"jackknife"‘. See the section on jackknife availability.
`transform`	One of ‘"none"‘, ‘"fisher"‘, or ‘"log_complement"‘. Wald uses a delta-method standard error on the selected scale; jackknife transforms the exact delete-one estimates directly.
`...`	Unused, present for S3 generic conformance.

Value

A two-column numeric matrix of lower and upper limits, one row per coefficient.

Jackknife availability

The exact delete-one summaries behind ‘method = "jackknife"‘ are attached automatically to fits with at least three subjects in these cases:

- [agreement()] on complete scalar ratings: ‘"ipw"‘ and ‘"cat_fiml"‘ (‘g = 2‘), and ‘"pairwise"‘.
- [agreement_counts()]: ‘"pairwise"‘ for any row totals, and ‘"cat_fiml"‘ when every row has the same total.

Fits on incomplete data, ‘g > 2‘, vector-valued ratings, the compound-symmetry estimators, and categorical FIML with ‘flatten > 0‘ support only ‘method = "wald"‘.

References

Efron, B., and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman and Hall, Chapter 11.

Clopper, C. J., and 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

```
fit <- agreement(dat.gwet2014, estimator = "ipw")
confint(fit)                        # natural-scale Wald interval
confint(fit, transform = "fisher") # Fisher z interval, always within (-1, 1)
complete_fit <- agreement_counts(dat.fleiss1971)
confint(complete_fit, method = "jackknife", transform = "fisher")
```

---

dat.fleiss1971 *Fleiss (1971) psychiatric diagnoses dataset*

Description

Counts-format rating data from Fleiss (1971), n = 30 subjects, six raters, five categories. Each row is one subject, and each column counts the number of raters who assigned that subject to that category.

Usage

```
dat.fleiss1971
```

Format

A 30 x 5 integer matrix.

Source

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5), 378-382.

Examples

```
agreement_counts(dat.fleiss1971, estimator = "pairwise")
```

dat.gwet2014 *Gwet (2014) example ratings with missing data*

Description

Raw rating data used as an example in Gwet (2014) and carried over from the legacy ‘misskappa‘ package: 20 subjects, five raters, four categories (‘0‘–‘3‘). Some ratings are missing (‘NA‘), so it is a compact worked example for the missing-data categorical estimators.

Usage

```
dat.gwet2014
```

Format

A 20 x 5 data frame. Columns ‘rater1‘–‘rater5‘ hold category codes ‘0‘–‘3‘, with ‘NA‘ marking unobserved ratings.

Source

Carried over from the legacy ‘misskappa‘ package. It appears as an example in Gwet, K. L. (2014), *Handbook of Inter-Rater Reliability*.

Examples

```
agreement(dat.gwet2014, estimator = "ipw", weight = "linear")
```

dat.klein2018 *Klein (2018) example ratings with missing data*

Description

Raw rating data carried over from the legacy ‘misskappa‘ package: 10 subjects, five raters, three categories (‘1‘–‘3‘), with some ratings missing (‘NA‘). A minimal example for the missing-data categorical estimators.

Usage

```
dat.klein2018
```

Format

A 10 x 5 data frame. Columns ‘rater1‘–‘rater5‘ hold category codes ‘1‘–‘3‘, with ‘NA‘ marking unobserved ratings.

Source

Carried over from the legacy 'misskappa' package (Klein, 2018).

References

Klein, D. (2018). Implementing a general framework for assessing interrater agreement in Stata. *The Stata Journal*, 18(4), 871-901. doi:10.1177/1536867X1801800408

Examples

```
agreement(dat.klein2018, estimator = "ipw")
```

---

dat.mcduff2019 *McDuff & Girard (2019) smiling-images agreement study*

Description

Agreement study from McDuff & Girard (2019) in long format: MTurk judges rate images of people smiling. The design is large and incomplete (many items and judges, and no judge rates every image), which makes it a useful missing-data example. There are 273 items, 121 judges, and 1365 rows.

Usage

```
dat.mcduff2019
```

Format

A 1365 x 4 data frame with one row per (item, judge) rating:

rating_positive numeric, the rated positivity of the image.

rating_smile numeric, the rated presence of a smile in the image.

item integer, the image rated.

judge integer, the MTurk judge giving the rating.

Source

The file `04_MTurk.csv` at https://osf.io/n4grd/.

References

McDuff, D., & Girard, J. M. (2019). Democratizing Psychological Insights from Analysis of Nonverbal Behavior. *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, 220-226. doi:10.1109/ACII.2019.8925503

Examples

```
# Each image is rated on a 1-6 smile scale by 5 of the 121 MTurk judges, and
# almost no two judges share images, so the judges are not individually
# identified. This is a crowdsourced design with exchangeable raters.
# Reshape to an (items x judges) matrix, collapse to per-image category
# counts with ratings_to_counts(), then estimate the scored (quadratically
# weighted) agreement.
smile <- with(dat.mcduff2019,
              tapply(rating_smile, list(item, judge), function(z) z[1]))
smile <- matrix(as.numeric(smile), nrow = nrow(smile))
agreement_counts(ratings_to_counts(smile), weight = "quadratic")
```

---

dat.tenhove2025 *Synthetic planned-incomplete ratings (compound-symmetry illustration)*

Description

A small *synthetic* inter-rater dataset built to illustrate the compound-symmetry (variance-component) estimators. It is **not real data**: 60 subjects are scored on an ordinal '1'–'5' scale by 6 raters drawn from a fixed-rater variance-component model 'y_ij = mu_j + b_i + e_ij', in the style of the *planned-incomplete* ICC designs of ten Hove, Jorgensen, and van der Ark (2025). Each subject is rated by only three of the six raters, in three overlapping panels (raters '1,2,3', then '3,4,5', then '5,6,1'), so the rater co-observation graph is *connected but not complete*. Some rater pairs (for example raters 2 and 4) never rate the same subject. That is exactly the regime where the saturated coefficients are not identified: 'agreement()' with 'estimator = "pairwise"', '"nt_fiml"', or '"cat_fiml"' stops with an identifiability error, while the compound-symmetry estimator '"nt_fiml_cs"', which needs only a connected graph, still fits. The build script 'data_raw/tenhove2025.R' regenerates it from a fixed seed.

Usage

```
dat.tenhove2025
```

Format

A 60 x 6 integer matrix. Columns 'rater1'–'rater6' hold ordinal scores '1'–'5', with 'NA' marking the (planned) unobserved ratings. Every row has exactly three observed entries.

Source

Simulated. Inspired by the planned-incomplete ICC designs of ten Hove et al. (2025). No real measurements are used.

References

ten Hove, D., Jorgensen, T. D., & van der Ark, L. A. (2025). How to Estimate ICCs for Interrater Reliability from Planned Incomplete Data. *Multivariate Behavioral Research*, 60(5). doi:10.1080/00273171.2025.2507745

Examples

```
# The co-observation graph is connected but incomplete, so the saturated
# estimators are not identified ...
try(agreement(dat.tenhove2025, estimator = "nt_fiml"))

# ... while the compound-symmetry estimator needs only connectivity.
agreement(dat.tenhove2025, estimator = "nt_fiml_cs", values = 1:5)
```

`dat.vanbelle2019` *Vanbelle (2019) crackles lung-sound agreement study (vector-valued ratings)*

Description

The CRACKLES auscultation study: 20 patients from the Tromso cohort, each recorded at six chest sites, classified for the presence of crackles (‘0‘/‘1‘) by 28 observers from seven groups of four (international experts ‘EXP‘, general practitioners from Norway ‘NOR‘, Russia ‘RUS‘, Wales ‘WAL‘ and the Netherlands ‘NLD‘, pulmonologists ‘PUL‘, and medical students ‘STU‘). Because each observer rates a patient at all six sites, every rating is a six-component *vector*, which makes this the worked example for the vector-valued (component-separable) path of [agreement()]: pass the array and pick a component loss with ‘weight‘. Ratings are complete (no missing entries). The data are factual binary classifications and are treated as public domain.

Usage

```
dat.vanbelle2019
```

Format

A 20 x 28 x 6 numeric array of ‘0‘/‘1‘ crackle classifications with named dimensions:

patient 20 patients (‘"1"‘–‘"20"‘).

rater 28 observers, four per group: ‘EXP1‘–‘EXP4‘, ‘NOR1‘–‘NOR4‘, ‘RUS1‘–‘RUS4‘, ‘WAL1‘–‘WAL4‘, ‘NLD1‘–‘NLD4‘, ‘PUL1‘–‘PUL4‘, ‘STU1‘–‘STU4‘.

site six chest sites: ‘U1‘, ‘U2‘ (upper posterior, left/right), ‘L1‘, ‘L2‘ (lower posterior), ‘A1‘, ‘A2‘ (anterior).

Source

The ‘CRACKLES‘ data frame distributed with the **multiagree** package (Vanbelle), reshaped into a subjects-by-raters-by-features array.

References

Vanbelle, S. (2019). Asymptotic variability of (multilevel) multirater kappa coefficients. *Statistical Methods in Medical Research*, 28(10-11), 3012-3026. doi:10.1177/0962280218794733

Aviles-Solis, J. C., Vanbelle, S., Halvorsen, P. A., et al. (2017). International perception of lung sounds: a comparison of classification across some European borders. *BMJ Open Respiratory Research*, 4(1), e000250. doi:10.1136/bmjresp2017000250

Examples

```
# Vector-valued agreement across the six sites for the four expert observers,
# with the default Hamming component loss (count of disagreeing sites).
agreement(dat.vanbelle2019[, 1:4, ], estimator = "pairwise")
```

`dat.zapf2016` *Zapf (2016) example ratings*

Description

Complete raw rating data carried over from the legacy 'misskappa' package: 50 subjects rated by four raters on a five-point scale ('1'–'5'). Having no missing entries, it is a convenient base for the complete-data coefficients and for illustrating the equality tests across rater pairs.

Usage

```
dat.zapf2016
```

Format

A 50 x 4 data frame. Columns 'Rater A'–'Rater D' hold category codes '1'–'5'.

Source

Carried over from the legacy 'misskappa' package (Zapf et al., 2016).

Examples

```
agreement(dat.zapf2016, estimator = "pairwise")
```

`influence.misskappa_estimate`
Per-subject influence functions for misskappa estimates

Description

'stats::influence()' returns the same per-subject influence-function matrix stored in 'fit$psi'. Rows are subjects and columns are coefficients, with scaling

$$\mathrm{vcov}(\hat{\theta}) = n^{-2}\Psi^{\top}\Psi.$$

This is the public S3 adapter for R's 'stats::influence()' generic. The 'fit$psi' component is what [agreement_test()] uses for paired tests.

Usage

```
## S3 method for class 'misskappa_estimate'
influence(model, ...)
```

Arguments

model	A ‘misskappa_estimate‘ object.
...	Unused. Present for S3 generic conformance.

Value

A numeric matrix of per-subject influence-function rows.

Examples

```
fit <- agreement(dat.gwet2014, estimator = "ipw")
psi <- stats::influence(fit)
all.equal(vcov(fit), crossprod(psi) / nrow(psi)^2)
```

---

ratings_to_counts *Convert rater-identified ratings to counts format*

Description

Turns a subjects-by-raters matrix of ratings into the subjects-by-categories counts used by [agreement_counts()]. Missing ratings are not counted, so a subject's row total is the number of raters who rated it. This discards rater identity: the raters become exchangeable.

Usage

```
ratings_to_counts(x, categories = NULL)
```

Arguments

x	A subjects-by-raters matrix or data frame of category labels (integer codes, characters, or factors). ‘NA‘ (and, for numeric input, any non-finite value) marks a missing rating.
categories	Optional vector of all categories, in column order. Unused categories become all-zero columns. Defaults to the shared factor levels, or else the sorted observed labels. Pass it for text ratings that have an order, since the column order sets the linear and quadratic weights in [agreement_counts()].

Value

An integer subjects-by-categories matrix whose columns are named by ‘categories‘ and whose row names are carried over from ‘x‘. Suitable as the ‘x‘ argument of [agreement_counts()].

Examples

```
counts <- ratings_to_counts(dat.gwet2014)
head(counts)
agreement_counts(counts)
```

Index

∗ **datasets**
dat.fleiss1971, 10
dat.gwet2014, 11
dat.klein2018, 11
dat.mcduff2019, 12
dat.tenhove2025, 13
dat.vanbelle2019, 14
dat.zapf2016, 15

agreement, 2
agreement_counts, 5
agreement_test, 7
as.data.frame.misskappa_estimate, 8

confint.misskappa_estimate, 8

dat.fleiss1971, 10
dat.gwet2014, 11
dat.klein2018, 11
dat.mcduff2019, 12
dat.tenhove2025, 13
dat.vanbelle2019, 14
dat.zapf2016, 15

influence.misskappa_estimate, 15

ratings_to_counts, 16