Package: grassr (via r-universe)

July 22, 2026

Type Package

Title Context-Conditioned Reporting for Binary Rater Reliability

Version 0.6.2

Description Generates a Report Card for rater reliability on binary outcomes from an N x k subject-by-rater rating matrix, on both the inter-rater and intra-rater axes. Each panel coefficient (PABAK, Gwet's AC1, Fleiss' kappa, and observed intraclass correlation) is positioned on a simulation-calibrated reference surface conditioned on the study's rater count, sample size, and prevalence, yielding a surface-position percentile with a bootstrap confidence qualifier. A cross-coefficient discordance diagnostic (delta-hat), with per-design thresholds calibrated to fixed false-positive rates, flags panels for which no single coefficient is a stable summary; for such divergent panels the report routes to a pairwise PABAK matrix and per-rater sensitivity and specificity recovered from a Dawid-Skene latent-class fit (Hui-Walter bounds at k = 2).

License MIT + file LICENSE

Encoding UTF-8

LazyData false

Depends R (>= 4.0)

Imports stats

Suggests boot, ggplot2, broom, future, future.apply, irr, irrCAC, patchwork, progressr, knitr, lme4 (>= 1.1-30), rmarkdown, testthat (>= 3.0.0), viridisLite

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

URL https://defense031.github.io/grassr/, https://github.com/defense031/grassr

BugReports https://github.com/defense031/grassr/issues

Repository https://defense031.r-universe.dev

Date/Publication 2026-07-09 19:11:02 UTC

RemoteUrl https://github.com/defense031/grassr

RemoteRef HEAD

RemoteSha d66f4b38f51acc6967574c9ccf12444ea0019639

Contents

as.data.frame.grass_asymmetry

Coerce a `grass_asymmetry` *result to a one-row data.frame*

Description

Coerce a `grass_asymmetry` result to a one-row data.frame

Usage

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

Arguments

`x`	A `grass_asymmetry` object.
`row.names, optional, ...`	Standard arguments; ignored except `row.names`.

Value

A one-row data.frame with `n_raters`, `delta_hat`, `summary`, `tier`, and `regime` columns.

as.data.frame.grass_card

Coerce a grass_card to a tidy data.frame (panel; per_rater appended on divergent)

Description

Returns the panel of observed coefficients with surface position metadata as a data.frame. The primary coefficient is flagged via the `is_primary` column. When the cross-coefficient flag is `divergent` and the per-rater latent-class table is populated, returns a list `c(panel = ..., per_rater = ...)` so tidy consumers can pivot the per-rater rows separately.

Usage

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

Arguments

`x`	A `grass_card` object.
`row.names, optional`	Standard `as.data.frame` arguments.
`...`	Unused.

Value

A `data.frame` (the panel) when `x$per_rater` is `NULL` or empty. When the divergent flag fires, returns a list with `panel` and `per_rater` data frames.

---

as.data.frame.grass_classification

Coerce a `grass_classification` *result to a one-row data.frame*

Description

Coerce a `grass_classification` result to a one-row data.frame

Usage

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

Arguments

`x`	A `grass_classification` object.
`row.names`, `optional`, `...`	Standard arguments; ignored except `row.names`.

Value

A one-row data.frame suitable for binding into a Reporting Card table.

---

as.data.frame.grass_surface_position

Coerce a grass_surface_position to a one-row data.frame

Description

Coerce a grass_surface_position to a one-row data.frame

Usage

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

Arguments

`x`	A `grass_surface_position` object.
`row.names`, `optional`, `...`	Standard arguments; ignored except `row.names`.

Value

A one-row data.frame summarising the position, label, and confidence qualifier. Band probabilities are flattened into columns p_band_1 ... p_band_4.

`check_asymmetry` *Cross-coefficient panel asymmetry diagnostic*

Description

`check_asymmetry()` takes an N x k binary ratings matrix and returns a scalar `delta_hat` (in percentile points, "pp"): the max-min spread of surface percentiles across the four agreement coefficients (PABAK, AC1, Fleiss kappa, Krippendorff alpha) when each is positioned on its calibrated reference surface. ICC is reported on the panel rows but excluded from `delta_hat` (its reference is distribution-sensitive in ways the agreement family is not).

Usage

```
check_asymmetry(
  ratings,
  axis = c("inter", "intra"),
  occasion = NULL,
  delta_thresholds = NULL,
  ...
)
```

Arguments

`ratings`	User input: an `N x k` binary matrix, an `N` x `k` data.frame whose columns are 0/1 / logical / 2-level factor, or a list of two equal-length 0/1 vectors (k = 2 paired form). See `?normalize_ratings` for accepted shapes.
`axis`	`"inter"` (default) or `"intra"`. Selects the surface family.
`occasion`	Reserved for `axis = "intra"` (a vector / factor identifying viewing occasion); ignored when `axis = "inter"`.
`delta_thresholds`	Length-2 numeric vector `c(caution, divergent)` in percentile points. Default `c(9.25, 11.75)` from paper Sec.3.2 (NP optimum at alpha = 0.05 / 0.01).
`...`	Forwarded to `position_on_surface()` (e.g. `bands`, `band_labels`, `reference_type`).

Details

`delta_hat` is a *split-bias* detector. It fires when raters tilt in different directions across (Se, Sp) – e.g., one rater high-Se / low-Sp, another high-Sp / low-Se – because the four coefficients respond to heterogeneous per-rater behavior differently and end up disagreeing on where the panel sits on the surface. The framework's other failure mode, *shared/uniform bias* (every rater tilts the same direction, e.g., a panel trained on one protocol all favoring specificity over sensitivity), produces

uniform percentile degradation across coefficients: small `delta_hat`, low minimum surface percentile. Shared bias is detected by the panel's minimum surface percentile, not by `delta_hat`. A divergent flag therefore identifies a specific kind of disagreement – cross-coefficient inversion from heterogeneous rater behavior – and routes the user to the per-rater pairwise PABAK matrix and pooled-reference (Se_tilde, Sp_tilde) diagnostic.

Each coefficient is positioned on its reference surface via `position_on_surface()`, yielding a percentile in `[0, 100]` pp. The panel diagnostic is `delta_hat = max(percentile) - min(percentile)`, computed over coefficients whose observed value sits within the achievable range of their reference surface (see *Surface-envelope clamp* below).

Value

An S3 object of class `grass_asymmetry_panel` with fields:

- `delta_hat`: scalar percentile-spread in pp
- `flag`: one of `"aligned"`, `"caution"`, `"divergent"`
- `thresholds`: named numeric vector of the (caution, divergent) cutoffs, in pp
- `panel`: data.frame with `coefficient`, `observed`, `percentile_pp`
- `notes`: character vector of unique caveats from the underlying surface positioning calls (e.g. nearest-neighbor gaps, ICC unavailability)

Surface-envelope clamp (v0.2.1+)

If an observed coefficient value falls outside the achievable range of its reference surface at the study's design `(pi_hat, k, N)`, the inversion to `q_hat` clamps to the boundary and the percentile lands at exactly `0` or `100`. Including such clamped percentiles in the max-min `delta_hat` would inflate the panel-spread purely because of the clamp, not because the panel disagrees on percentile. Since v0.2.1 the function therefore *excludes* clamped coefficients from `delta_hat` whenever at least two unclamped coefficients remain. The affected coefficients are still shown in the returned `panel` data.frame with `clamped = TRUE` and surface percentile `0` or `100`, and a note in `$notes` names which coefficients were excluded. This matters most often for ICC at `N > 200` (the bundled fitted-ICC reference is calibrated through `N = 200`; beyond that the function falls back to the oracle reference, which over-predicts the practitioner's GLMM-fit ICC and therefore clamps). The behavior guarantees that a `divergent` flag always reflects genuine cross-coefficient surface-percentile disagreement rather than an envelope artifact; the trade-off is that a coefficient with no calibrated reference at the user's design contributes nothing to the diagnostic. If fewer than two unclamped coefficients remain, `delta_hat` falls back to the raw spread including clamped values and the note records the fallback.

Tier thresholds default to `c(9.25, 11.75)` pp, the Neyman-Pearson-motivated size-alpha cutoffs at `alpha = 0.05` and `alpha = 0.01` from Sec.3.2 of the GRASS paper (see also App G operating characteristics). The construction is NP-optimal within the test class of threshold rules on `delta_hat`; it is not a likelihood-ratio test on the joint distribution of the panel coefficients (`delta_hat` is a paired-margin difference, not a likelihood ratio). The three flags are:

- `aligned` (`delta_hat < 9.25` pp): the panel agrees on the surface position. Any single coefficient is a stable summary; the primary coefficient (Table 2) carries the headline.
- `caution` (`9.25 <= delta_hat < 11.75` pp): the panel is mildly inconsistent. Report the primary coefficient with a caution flag and the `delta_hat` value.

- `divergent` (`delta_hat >= 11.75` pp): no single coefficient is a stable summary. Use `latent_class_fit()` to recover per-rater `(Se, Sp)` and report those instead.

The new `check_asymmetry(ratings, ...)` signature replaces the v0.1.x `check_asymmetry(se, sp, ...)` per-rater signature. For backward compatibility, calling `check_asymmetry()` with `se = ...` and `sp = ...` named arguments emits a one-time deprecation hint and routes the call to `check_rater_asymmetry()`. Supplying both `ratings = ...` and per-rater `se =` / `sp =` is an error.

References

Neyman, J. and Pearson, E. S. (1933). On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London A*, 231, 289-337.

See Also

`check_rater_asymmetry()` for the per-rater Se/Sp companion; `latent_class_fit()` for the divergent-branch recovery of per-rater `(Se, Sp)`; `position_on_surface()` for the underlying surface positioning.

Examples

```
set.seed(1)
# Build a 5x200 symmetric panel -- should print as 'aligned'.
Y <- matrix(rbinom(5 * 200, 1, 0.30), nrow = 200, ncol = 5)
check_asymmetry(Y)
```

`check_rater_asymmetry` *Per-rater Se/Sp asymmetry diagnostic*

Description

`check_rater_asymmetry()` is the rater-level companion to `check_asymmetry()`. It takes per-rater sensitivity and specificity estimates, computes the per-rater gap `|Se_j - Sp_j|`, reduces them to a scalar `delta_hat = mean_norm |Se_j - Sp_j|` (or `max`), and assigns the three-tier model-safety regime that governs whether `q_hat` (the GRASS operating-quality projection onto the Se = Sp diagonal) is trustworthy as the primary summary.

Usage

```
check_rater_asymmetry(
  se,
  sp,
  rater = NULL,
  threshold_caution = 0.05,
  threshold_unsafe = 0.1,
  summary = c("max", "mean")
)
```

Arguments

se	Numeric vector, one per rater. Per-rater sensitivity estimates in [0, 1].
sp	Numeric vector, one per rater. Per-rater specificity estimates in [0, 1]. Same length as se.
rater	Optional character vector of rater labels, same length as se. Defaults to "R1", "R2", ...
threshold_caution	Boundary between Tier 1 (ok) and Tier 2 (caution). Default 0.05.
threshold_unsafe	Boundary between Tier 2 (caution) and Tier 3 (unsafe). Default 0.10.
summary	How to reduce per-rater gaps to the scalar delta_hat. "max" (default, conservative tripwire – any single rater above the threshold triggers escalation) or "mean" (panel-average asymmetry). The framework uses "max"; "mean" is provided for sensitivity checks.

Details

This function is appropriate when the user already has per-rater Se/Sp estimates – typically from a Hui-Walter / Dawid-Skene latent-class fit (see latent_class_fit()), a simulation with known truth, or a reference-standard comparison. It is the function called inside the divergent branch of the new check_asymmetry() / grass_report() workflow when per-rater (Se, Sp) become available.

For ratings-matrix input (when only the N x k binary ratings are available, no per-rater Se/Sp), use check_asymmetry() instead – it computes the cross-coefficient percentile spread without requiring identifiability of per-rater (Se, Sp).

Tier thresholds follow the three-tier architecture:

- **Tier 1 –** ok (delta_hat < 0.05): diagonal default. Report q_hat +/- SE plus EDR and EMR_panel without per-rater disclosure.
- **Tier 2 –** caution (0.05 <= delta_hat < 0.10): diagonal + diagnostic. Report q_hat +/- SE with a caution flag and the delta_hat value.
- **Tier 3 –** unsafe (delta_hat >= 0.10): full latent-class. q_hat is withheld as primary; report per-rater (Se_j, Sp_j) via a Hui-Walter / Dawid-Skene fit. PABAK's prevalence-flatness was conditional on Se = Sp; at Tier 3 that condition is visibly broken.

se and sp are not identifiable from a single 2-rater binary table without external ground truth. Supply them from: (a) a simulation with known truth, (b) a Hui-Walter / Dawid-Skene latent-class fit under k >= 3 and prevalence heterogeneity, or (c) a reference-standard comparison. Passing raw table() off-diagonals as se / sp is **wrong** and the function cannot detect the error.

Value

An S3 object of class grass_asymmetry with fields:

- per_rater: data.frame of rater, se, sp, and gap = |se - sp|
- delta_hat: the scalar delta_hat summary
- summary: which summary statistic was used ("max" or "mean")

- `tier`: integer tier (1, 2, or 3)
- `regime`: character regime label (`"ok"`, `"caution"`, or `"unsafe"`)
- `thresholds`: named list of the caution and unsafe cutoffs

See Also

`check_asymmetry()` for the ratings-matrix companion (panel-level percentile spread). `latent_class_fit()` for the recommended source of `(se, sp)` estimates.

Examples

```
# Tier 1: symmetric raters -- safe to use q_hat as primary
check_rater_asymmetry(se = c(0.86, 0.88, 0.84), sp = c(0.85, 0.87, 0.86))

# Tier 2: one rater pressing the Se = Sp assumption
check_rater_asymmetry(se = c(0.90, 0.88, 0.92), sp = c(0.82, 0.86, 0.85))

# Tier 3: within-rater Se-favoring regime -- requires latent-class fit
check_rater_asymmetry(se = c(0.95, 0.93, 0.94), sp = c(0.78, 0.80, 0.79))
```

`format.grass_card` *Format a grass_card as a character vector*

Description

Returns the four-field Report Card rendered as a character vector, one line per element. Used by `print.grass_card()` and by downstream embedders (Markdown, knitr) that want the lines as data.

Usage

```
## S3 method for class 'grass_card'
format(x, digits = 2, ...)
```

Arguments

`x`	A `grass_card` object (returned by `grass_report()`).
`digits`	Numeric. Decimals for observed coefficient values. Default 2.
`...`	Ignored.

Value

A character vector (one element per line).

`grass_compute` *Compute the full inter-rater agreement panel*

Description

Computes Cohen's kappa, PABAK, Gwet's AC1, positive and negative agreement, Byrt prevalence and bias indices, and two confidence intervals for kappa (Wald, and a logit-transformed Wilson-type interval).

Usage

```
grass_compute(
  data,
  format = c("wide", "matrix", "long", "paired"),
  positive = NULL,
  ...
)
```

Arguments

`data`	Input data. Form depends on `format`: • `"matrix"` – a 2x2 integer count matrix with `R1` rows, `R2` columns. • `"wide"` – a data.frame with exactly two rater columns (or pass `rater_cols = c("r1", "r2")` if the data contains extra columns). • `"long"` – a data.frame with `subject`, `rater`, and `rating` columns (names configurable via `subject =`, `rater =`, `rating =`). Must contain exactly two distinct raters. • `"paired"` – a list of two equal-length rating vectors, a 2-column matrix, or a 2-column data.frame.
`format`	One of `"matrix"`, `"wide"`, `"long"`, `"paired"`.
`positive`	Optional character string naming the level to treat as the positive (=1) class. Defaults to `"yes"` if present among levels, then `"1"` / `"true"` / `"positive"` / `"case"`, then first-encountered.
`...`	Format-specific arguments: `rater_cols` for `"wide"`; `subject`, `rater`, `rating` for `"long"`.

Value

A `grass_metrics` S3 object.

Deprecated

`grass_compute()` is deprecated in grass 0.2.0. The new headline API is `grass_report(ratings = Y)` returning a `grass_card` object. The full coefficient panel is available via `summary()` or `as.data.frame()`. See `vignette("reporting-card")` and `grass_report()`.

Examples

```
# 2x2 counts (Cohen 1960 Table 1)
tab <- matrix(c(88, 10, 14, 88), nrow = 2,
              dimnames = list(R1 = c("0", "1"), R2 = c("0", "1")))
grass_compute(tab, format = "matrix")

# Paired vectors
r1 <- c(1, 1, 0, 0, 1, 0, 1)
r2 <- c(1, 0, 0, 0, 1, 1, 1)
grass_compute(list(r1, r2), format = "paired")
```

---

grass_plot *Plot a grass object*

Description

Dispatches to `plot.grass_result` or `plot.grass_reference`.

Usage

```
grass_plot(x, ...)
```

Arguments

x	A grass object.
...	Passed through to the class-specific plot method.

Value

A ggplot object.

---

grass_prevalence *Estimate prevalence of the positive class from rater data*

Description

Averages the marginal positive rates of the two raters.

Usage

```
grass_prevalence(
  data,
  format = c("wide", "matrix", "long", "paired"),
  positive = NULL,
  ...
)
```

Arguments

data	Input data. Form depends on format: • "matrix" – a 2x2 integer count matrix with R1 rows, R2 columns. • "wide" – a data.frame with exactly two rater columns (or pass rater_cols = c("r1", "r2") if the data contains extra columns). • "long" – a data.frame with subject, rater, and rating columns (names configurable via subject =, rater =, rating =). Must contain exactly two distinct raters. • "paired" – a list of two equal-length rating vectors, a 2-column matrix, or a 2-column data.frame.
format	One of "matrix", "wide", "long", "paired".
positive	Optional character string naming the level to treat as the positive (=1) class. Defaults to "yes" if present among levels, then "1" / "true" / "positive" / "case", then first-encountered.
...	Format-specific arguments: rater_cols for "wide"; subject, rater, rating for "long".

Value

A single numeric in [0, 1].

Examples

```
tab <- matrix(c(88, 10, 14, 88), nrow = 2,
              dimnames = list(R1 = c("0", "1"), R2 = c("0", "1")))
grass_prevalence(tab, format = "matrix")
```

---

grass_reference *Look up prevalence-conditioned reference values for agreement metrics*

Description

Returns the prevalence-conditioned reference values for Cohen's kappa, PABAK, and Gwet's AC1 at a given prevalence and a chosen rater-quality band. The reference is the analytical Youden-J-optimal metric value on the Se = Sp diagonal at the chosen reference_level, under conditional independence of the two raters given the true class.

Usage

```
grass_reference(prevalence, reference_level = 0.85, quality = NULL)
```

Arguments

prevalence	Numeric in [0, 1]. Values outside [0.01, 0.99] are clamped with a warning.
reference_level	One of 0.70, 0.80, 0.85, 0.90. Default 0.85.
quality	Deprecated alias. "high" -> reference_level = 0.85, "medium" -> 0.70. Emits a one-time soft deprecation warning.

Details

In the paper these are called the *prevalence-conditioned thresholds*. The package uses the name reference because its API reports the values alongside prevalence index and bias index rather than applying them as cutoffs.

Value

A grass_reference S3 object containing a 3-row data.frame (one row per metric) with columns metric, reference, J, reference_level, quality, prevalence. The quality column retains the legacy "high"/"medium" label for 0.85 and 0.70 bands and is NA_character_ for the new 0.80 / 0.90 bands.

Deprecated

grass_reference() is deprecated in grass 0.2.0. The percentile machinery in position_on_surface() and the new grass_report() supersede the fixed reference-curve lookup. See vignette("reporting-card") for the v0.2.0 surface-position workflow.

Examples

```
grass_reference(0.5)
grass_reference(0.1, reference_level = 0.70)
```

---

grass_reference_table *Return the full internal reference table*

Description

For inspection, custom interpolation, or reproducibility. In the paper this is the prevalence-conditioned threshold table.

Usage

```
grass_reference_table(reference_level = NULL)
```

Arguments

reference_level	Optional numeric band (0.70, 0.80, 0.85, or 0.90). If supplied, only that band is returned. Default NULL returns all four bands in long form.

Value

A long-form data.frame with columns `prevalence`, `reference_level`, `metric`, `reference`, and `J`.

Examples

```
head(grass_reference_table())
head(grass_reference_table(reference_level = 0.85))
```

`grass_report` *Generate a GRASS Report Card from a rating matrix*

Description

`grass_report()` is the headline entry point for the v0.2.0 Target-2 framework. It takes an `N x k` binary rating matrix and returns a four-field Report Card: the sample summary (`k, N, pi_hat`), the primary coefficient and its surface position, the cross-coefficient asymmetry diagnostic `delta_hat` and flag, and (when `flag == "divergent"`) the per-rater latent-class fit. The full panel of coefficients, surface percentiles, band probabilities, and reference-surface artifacts ride along on the same object for `summary()`, `as.data.frame()`, and `plot()` access.

Usage

```
grass_report(
  ratings,
  axis = c("inter", "intra"),
  metric = "auto",
  occasion = NULL,
  bands = c(0.5, 0.625, 0.75, 0.875, 1),
  band_labels = c("Poor", "Moderate", "Strong", "Excellent"),
  delta_thresholds = NULL,
  bootstrap_B = 1000L,
  bootstrap_delta_B = 0L,
  verbose = FALSE,
  ...
)
```

Arguments

`ratings`	User input: an `N x k` binary matrix, an `N x k` data.frame whose columns are 0/1 / logical / 2-level factor, or a list of two equal-length 0/1 vectors (`k = 2` paired form). See `?normalize_ratings` for accepted shapes.
`axis`	One of `"inter"` (default) or `"intra"`. Selects the surface family. The intra-axis path uses `occasion` to identify viewings.

metric	One of "auto" (default; calls pick_primary_coefficient() per Table 2), "pabak", "ac1", "fleiss_kappa", "icc". Selects which coefficient is the headline in the printed Report Card; the full panel is always populated. (Krippendorff's alpha left the Report Card panel at v0.6.0; it coincides with Fleiss' kappa in the binary fully-crossed case. Use obs_krippendorff_alpha() or position_on_surface() to compute it manually.)
occasion	Reserved for axis = "intra"; ignored when axis = "inter".
bands	Numeric length-5 partition on q in [0.5, 1]. Default c(0.5, 0.625, 0.75, 0.875, 1.0).
band_labels	Character length-4 labels for the bands. Default c("Poor", "Moderate", "Strong", "Excellent").
delta_thresholds	Length-2 numeric vector c(caution, divergent) in percentile points. Default c(9.25, 11.75) (paper Sec.3.2, NP-motivated size-alpha).
bootstrap_B	Integer; bootstrap replicates for the divergent-branch latent-class CIs. Default 1000L. Set lower for fast tests.
bootstrap_delta_B	Integer; subject-resampling replicates for the optional bootstrap distribution of delta_hat. Default 0L (off); values below 50L are treated as off.
verbose	Logical; emit progress messages on long calls. Default FALSE.
...	Reserved for future extension.

Details

The body, in order:

1. Normalize ratings to a canonical N x k integer matrix Y and derive pi_hat = mean(Y), k = ncol(Y), N = nrow(Y). Validate k >= 2; warn at N < 10; note at N < 30.
2. Compute the panel of observed coefficients (compute_panel(), internal): at k = 2, PABAK / AC1 / Cohen's kappa; at k >= 3, PABAK / AC1 / Fleiss kappa / ICC.
3. For each panel coefficient, position the observed value on its DGP-calibrated reference surface via position_on_surface().
4. Pick the primary coefficient via Table 2 (metric = "auto") or accept the user's override.
5. Compute the cross-coefficient percentile spread delta_hat (in pp) via check_asymmetry() and tier into aligned / caution / divergent.
6. If flag == "divergent": run a latent_class_fit() (Dawid-Skene EM at k >= 3; Hui-Walter bounds at k = 2) and attach the per-rater (Se_j, Sp_j) table.
7. Assemble the grass_card S3 object.

Value

An object of class c("grass_card", "list") with fields sample, coefficient, delta, panel, per_rater, surface, call, grass_version, timestamp, inputs, notes. See the v0.2.0 paper-alignment design doc Sec.3.1 for the full structure.

Percentile units. card$coefficient$surface_percentile and card$panel$surface_percentile are reported on the 0-100 scale (e.g., 46.3 means the 46th percentile). The underlying position_on_surface()

returns `percentile` on the 0-1 fraction scale; `grass_report()` multiplies by 100 to match the paper's prose convention. The `print` and `format` methods use ordinal notation ("46th percentile").

Examples

```
set.seed(1)
Y <- matrix(rbinom(1000, 1, 0.3), nrow = 200, ncol = 5)
card <- grass_report(ratings = Y)
card                        # print
summary(card)               # full panel + per-rater
as.data.frame(card)         # tidy long-format
```

`grass_roadmap` *The grass ecosystem roadmap*

Description

`grass` is the binary categorical-agreement submodule of the MEADOW framework. Its single headline entry point – `grass_report()` – takes an `N x k` binary rating matrix and returns a `grass_card`: a four-field Report Card carrying the sample summary, primary coefficient with its surface-position percentile, the four-band qualitative label, and the cross-coefficient asymmetry diagnostic `delta_hat` with its three-tier stability flag. The full panel of coefficients, percentiles, bootstrap CIs, and reference-surface artifacts ride along on the same object for `summary()`, `as.data.frame()`, and `plot()` access.

Framework foundation

A core idea animates the framework: **fixed interpretation bands (Landis-Koch 1977 and its descendants) are mathematically invalid across prevalences, sample sizes, and rater counts.** The Target-2 principle of MEADOW is *context-conditioned reporting*: the percentile a coefficient lands at and the qualitative band that label maps to are both conditioned on the actual study `(k, N, pi_hat)`, computed by inverting a calibrated reference surface rather than read off a fixed table. `grass` ships the binary submodule; FIELD is planned to ship alongside Paper 3 with the same Target-2 contract for variance-component reliability on continuous outcomes.

MEADOW submodules

MEADOW is the umbrella; each submodule covers one scale type. The user-facing API of each submodule is the same – a single rating matrix in, a Report Card out – so user code that works on a binary panel today will work on a continuous panel once FIELD ships.

Submodule	Scope
GRASS	Binary categorical agreement (Cohen's kappa, PABAK, AC1, Fleiss kappa, Krippendorff alpha, observed ICC f
FIELD	Continuous variance-component reliability (Shrout-Fleiss ICC family, Lin's CCC, Bland-Altman bounds, gener

Earlier drafts of the roadmap referenced TURF and a separate MEADOW submodule for nominal multi-rater agreement. Those are retired: GRASS now covers the binary multi-rater case (Fleiss kappa, Krippendorff alpha, observed ICC) directly, and the framework taxonomy collapses to MEADOW = GRASS + FIELD.

Stable API

The Target-2 contract is the same across submodules:

- `grass_report()` – primary entry point; rating matrix in, `grass_card` out. The `grass_card` carries the four-field summary (sample, primary coefficient with surface percentile, four-band label, `delta_hat` with stability flag), the full panel of coefficients with their bootstrap CIs, and the reference-surface artifacts.
- `position_on_surface()` – granular access to a single coefficient's surface percentile, four-band label, and qualifier.
- `check_asymmetry()` – granular access to the cross-coefficient `delta_hat` and three-tier stability flag.
- `latent_class_fit()` – per-rater Se/Sp via Dawid-Skene EM (k >= 3) or Hui-Walter bounds (k = 2), populated automatically in the divergent branch of `grass_report()`.
- `summary()`, `as.data.frame()`, `plot()` – layered access to the full underlying panel and the surface-position visualization.

Target-2 vocabulary

These terms, used throughout the package documentation and printed Report Card, come from the merged GRASS binary-rater-reliability paper (Sec.Sec.3-4):

- **context-conditioned reporting convention** – the principle that the percentile and band reported for a coefficient must condition on `(k, N, pi_hat)`, not on a fixed table.
- **surface-position percentile** – the empirical percentile of the observed coefficient against the calibrated reference surface at the study's `(k, N, pi_hat)`.
- **four-band label** – the qualitative tier (`Poor` / `Moderate` / `Strong` / `Excellent`) mapped from `q_hat` (the operating-quality projection onto the `Se = Sp` diagonal) via the partition `c(0.5, 0.625, 0.75, 0.875, 1.0)`.
- **delta-hat (`delta_hat`) stability flag** – the cross-coefficient percentile spread (in pp) with three tiers `aligned` / `caution` / `divergent` at NP-motivated size-alpha thresholds `c(9.25, 11.75)` (paper Sec.3.2, App G operating characteristics). When `divergent`, the band is suppressed and per-rater Se/Sp from a latent-class fit are reported instead.

What is not yet implemented

Constructing one of the placeholder specs (`grass_spec_continuous()`, `grass_spec_multirater()`, `grass_spec_ordinal()`) is legal so users can write code ready for FIELD or for future GRASS extensions. Passing a placeholder spec to a dispatching function errors with a pointer back here. The placeholders are kept so that a paper or a user-side script that already names a future submodule by spec continues to parse.

Paper

The foundational paper for MEADOW and the GRASS submodule is in review (Semmel 202X, *Context-Conditioned Reporting for Binary Rater Reliability*). The FIELD paper will follow, citing the GRASS paper as the methodological precedent. Each paper accompanies a minor or major release of this package rather than a new package.

grass_spec_binary *Binary two-rater agreement spec*

Description

The spec for the currently-implemented metric family: Cohen's kappa, PABAK, and Gwet's AC1 for two raters on a binary outcome, evaluated against a prevalence-conditioned reference curve at the chosen `reference_level`.

Usage

```
grass_spec_binary(reference_level = 0.85)
```

Arguments

`reference_level`
: One of `0.70`, `0.80`, `0.85`, `0.90`. Selects the Se = Sp rater-quality band whose analytical Youden-J reference curve is attached to the result. `0.85` is the default and reproduces the "high" quality curve from prior releases; `0.70` reproduces the "medium" curve. `0.80` and `0.90` are new in 0.1.2. Pass `NULL` to skip the reference attachment entirely (metrics in context only).

Value

A `grass_spec_binary` object carrying `$family = "binary"` and `$reference_level`.

Examples

```
grass_spec_binary()
grass_spec_binary(reference_level = 0.80)
```

grass_spec_continuous *Continuous agreement spec (planned)*

Description

Placeholder for the continuous family: intraclass correlation coefficient (Shrout-Fleiss forms), Lin's concordance correlation coefficient, Bland-Altman limits of agreement (parametric and non-parametric), standard error of measurement, and generalizability theory variance components.

Usage

```
grass_spec_continuous(...)
```

Arguments

`...`
: Ignored in 0.1.2; reserved for future parameters.

Details

See ?grass_roadmap.

Value

A placeholder grass_spec_continuous object.

See Also

grass_roadmap

`grass_spec_multirater` *Multi-rater nominal agreement spec (planned)*

Description

Placeholder for Fleiss' kappa and nominal Krippendorff's alpha (more than two raters, nominal scale), with prevalence-conditioned reference curves analogous to the binary family.

Usage

```
grass_spec_multirater(...)
```

Arguments

...	Ignored in 0.1.2; reserved for future parameters.

Details

See ?grass_roadmap.

Value

A placeholder grass_spec_multirater object.

See Also

grass_roadmap

grass_spec_ordinal *Ordinal two-rater agreement spec (planned)*

Description

Placeholder for the GRASS ecosystem's ordinal family: weighted Cohen's kappa (linear and quadratic weights), Gwet's AC2, and ordinal Krippendorff's alpha, each with prevalence-conditioned reference curves analogous to the binary family.

Usage

```
grass_spec_ordinal(...)
```

Arguments

...	Ignored in 0.1.2; reserved for future parameters.

Details

Constructing the spec is legal (so users can write code that is ready for the release); passing it to `grass_report()` errors until the implementation lands. See `?grass_roadmap` for the framework taxonomy.

Value

A placeholder `grass_spec_ordinal` object.

See Also

grass_roadmap

latent_class_fit *Latent-class fit: per-rater Sensitivity and Specificity from a binary rating matrix*

Description

`latent_class_fit()` is the divergent-branch fallback for the GRASS Reporting Card. When the cross-coefficient panel disagrees, the framework abandons the single q_hat summary and reports per-rater (Se_j, Sp_j) instead. This function fits those per-rater accuracy parameters from an N x k binary rating matrix.

Usage

```
latent_class_fit(
  ratings,
  B = 1000L,
  method = NULL,
  max_iter = 1000L,
  tol = 1e-06,
  seed = NULL,
  ...
)
```

Arguments

ratings	An N x k binary rating matrix. Rows are subjects, columns are raters, values in {0, 1}. Data.frame and list-of-rater- columns inputs are coerced. Requires N >= 10, k >= 2, no NA, no all-constant columns.
B	Integer >= 0. Number of nonparametric bootstrap replicates. `B = 0` skips the bootstrap and returns the EM point estimate (or bounds at k = 2) without CIs.
method	One of `"dawid_skene_em"`, `"hui_walter"`, or `NULL` (default). When `NULL`, dispatches to Hui-Walter at k = 2 and to Dawid-Skene EM at k >= 3. Supplying `"dawid_skene_em"` at k = 2 errors – the EM is unidentified there.
max_iter	Integer. EM iteration cap. Default 1000.
tol	Numeric. EM log-likelihood tolerance. Default 1e-6.
seed	Optional integer. Sets the bootstrap RNG; the EM itself is deterministic.
...	Reserved for future extensions; currently ignored.

Details

Two regimes, dispatched by k:

- **k >= 3: Dawid-Skene 1979 expectation-maximization.** Latent binary class C_i in {0, 1}; conditional independence of raters given C; parameters (pi, Se_j, Sp_j) for j = 1..k. Initialized from majority-vote consensus. Iterated to log-likelihood tolerance `tol` or `max_iter`. Hand-rolled in base R, no new package dependency.
- **k == 2: Hui-Walter 1980 inequality bounds.** Per-rater Se and Sp are NOT point-identified from a single 2x2 table without external information (e.g., a known-prevalence subgroup or a reference standard). We return the inequality bounds the observed marginals support: with rater j positive rate `P_j`, `Se_j` is bounded below by `P_j`, `Sp_j` is bounded below by `1 - P_j`, both above by 1. The returned rows have `bound_only = TRUE` and `se_hat = sp_hat = NA`. This is a documented limitation of the design, not a bug.

Bootstrap CI (when `B > 0`): nonparametric, subjects-with-replacement. At k >= 3 the EM is refit on each bootstrap sample; at k = 2 the bounds are recomputed and the bootstrap distribution is on the bound midpoints. Reports the empirical 2.5 / 97.5 percentiles on the per-rater Se_j and Sp_j (or their bound midpoints at k = 2).

Value

An S3 object of class `c("grass_latent_class", "list")`:

- `per_rater`: data.frame with `rater`, `se_hat`, `sp_hat` (NA at k = 2), `se_lower`, `se_upper`, `sp_lower`, `sp_upper`, `bound_only` (TRUE at k = 2).
- `method`: `"dawid_skene_em"` or `"hui_walter"`.
- `converged`: logical (NA at k = 2).
- `iterations`: integer (NA at k = 2).
- `B`: integer; bootstrap replicates run.
- `prevalence_hat`: numeric estimated prevalence (NA at k = 2).
- `log_likelihood`: numeric (NA at k = 2).

References

Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28(1), 20–28.

Hui, S. L. and Walter, S. D. (1980). Estimating the error rates of diagnostic tests. *Biometrics*, 36(1), 167–171.

Examples

```
## Not run:
set.seed(1)
N <- 500; k <- 5
Se <- 0.90; Sp <- 0.85; pi <- 0.30
C <- rbinom(N, 1, pi)
Y <- matrix(0L, N, k)
for (j in seq_len(k))
  Y[, j] <- rbinom(N, 1, ifelse(C == 1, Se, 1 - Sp))
fit <- latent_class_fit(Y, B = 200, seed = 1)
print(fit)

## End(Not run)
```

---

obs_krippendorff_alpha

Krippendorff's alpha (nominal, binary, fully crossed)

Description

Fully-crossed design (`m_i` = k for all i). Hayes-Krippendorff coincidence- matrix reduction: `o_{01} = 2 * sum_i r_i * (k - n_0 = N*k - sum(r_i)`, `n_1 = sum(r_i)`, `n_total = N*k`, `alpha = 1 - (n_total - 1) * o_{01} / (2 * n_0 * n_1)`.

Usage

```
obs_krippendorff_alpha(Y)
```

Arguments

Y	N x k integer matrix in {0L, 1L}.

Details

Returns NA_real_ (with a note attribute) if the design is degenerate (no positives or no negatives in Y).

As of v0.6.0, alpha is not part of the Report Card panel or the delta_hat coefficient set: in the binary fully-crossed case it coincides with Fleiss' kappa asymptotically (median absolute difference 0.00024 across the 10,140-cell calibration grid). This function is exported for users who need the value for cross-study comparison.

Value

Single numeric.

---

pairwise_agreement *Pairwise reliability for a divergent panel*

Description

Computes pairwise PABAK between every pair of raters in an $N \times k$ binary rating matrix and places each entry on the $k = 2$ reference surface at the pair's observed marginal. Also returns per-rater pooled-reference sensitivity and specificity — each rater's call rate against the panel-majority of the other $k - 1$ raters — which uses the larger panel's information about rater behavior rather than discarding it to a strictly pairwise comparison.

Usage

```
pairwise_agreement(ratings, axis = c("inter", "intra"))
```

Arguments

ratings	An $N \times k$ binary rating matrix (rows = subjects, columns = raters), or a data.frame whose columns are 0/1 / logical / two-level factor. Same input conventions as grass_report.
axis	Character; "inter" (default) or "intra". The intra-axis path treats the columns of ratings as per-rater viewing pairs; see paper Sec.3.3 intra-rater section.

Details

This function is the recommended primary deliverable when grass_report() flags the panel as *divergent*; the panel-aggregate coefficients no longer summarize the panel adequately, but the pairwise matrix exposes the panel's structure directly (uniform inconsistency, sub-group clustering, or single- rater outliers).

Value

An object of class `c("grass_pairwise", "list")` with fields:

- `pabak_matrix` — $k \times k$ symmetric numeric matrix; $[i, j]$ is PABAK_{ij}, the diagonal is 1.
- `percentile_matrix` — $k \times k$ symmetric numeric (0–100); $[i, j]$ is the surface percentile of PABAK_{ij} on the $k = 2$ reference at the pair's observed marginal. Diagonal is `NA`.
- `marginal_matrix` — $k \times k$ symmetric numeric; $[i, j]$ is $\hat{\pi}_{+,ij}$. Diagonal is `NA`.
- `band_matrix` — $k \times k$ character matrix; four-band label for each pairwise percentile. Diagonal is `NA`.
- `qualifier_matrix` — $k \times k$ character; decisive / moderate / weak per pairwise percentile.
- `pooled_per_rater` — data frame with k rows, one per rater, columns `rater`, `se_tilde`, `sp_tilde`, `n_pool_pos`, `n_pool_neg`, `n_pool_excluded` (subjects with tied panel majority).
- `sample` — list with `k`, `N`, `pi_hat`, `tau2_hat`, `axis`.
- `notes` — character vector with caveats, if any (e.g., undefined pooled-reference at $k = 2$).
- `call` — the matched call.

Examples

```
set.seed(6)
Se <- c(0.95, 0.75, 0.95, 0.75, 0.95)
Sp <- c(0.75, 0.95, 0.75, 0.95, 0.75)
truth <- rbinom(200, 1, 0.5)
Y <- sapply(seq_along(Se),
            function(j) ifelse(truth == 1,
                               rbinom(200, 1, Se[j]),
                               rbinom(200, 1, 1 - Sp[j])))
pw <- pairwise_agreement(Y)
pw
```

---

plot.grass_card *Plot a* grass_card *Report Card object*

Description

Six views of a Report Card returned by `grass_report()`. The default `"surface"` view places the study on its DGP-calibrated reference surface; the other views are for cross-coefficient comparison, the asymmetry diagnostic, observed-coefficient confidence intervals, per-rater Se/Sp from the latent-class fit (only valid when `card$delta$flag == "divergent"`), and a `patchwork` composite of the most useful at-a-glance panels.

Usage

```
## S3 method for class 'grass_card'
plot(
  x,
  type = c("surface", "panel", "thermometer", "intervals", "per_rater", "pairwise",
    "diagnostic"),
  ...
)
```

Arguments

x	A grass_card object.
type	One of:
	• "surface" (default) – 2D heatmap of the expected primary metric over (M_1, q) at the study's (k, N), with the observation pinned and dotted band contours at q in {0.5, 0.625, 0.75, 0.875, 1.0}.
	• "panel" – forest plot of all panel coefficients on the percentile axis (0-100 pp), so the cross-coefficient spread is visible.
	• "thermometer" – colored gauge for delta_hat with the aligned/caution/divergent thresholds shown.
	• "intervals" – forest plot of observed coefficients with 95% Wilson-logit CIs.
	• "per_rater" – forest plot of per-rater Se-hat and Sp-hat (latent-class bootstrap CIs). Errors when card$per_rater is NULL.
	• "pairwise" – k x k tile heatmap of pairwise surface percentiles (cell label = percentile in pp; parenthetical label = PABAK_ij). Only available when card$delta$flag == "divergent" (auto-populated by grass_report() via pairwise_agreement()).
	• "diagnostic" – patchwork composite of "panel" + "thermometer" + "pairwise" (under divergent) / "per_rater" / "intervals".
...	Currently unused.

Value

A ggplot object (or a patchwork composite for type = "diagnostic").

---

plot.grass_result *Plot a grass report*

Description

The default "landing" plot overlays observed kappa, PABAK, and AC1 on the GRASS prevalence-conditioned reference curves. The "regime" plot places the study on the PI^2 vs BI^2 plane.

Usage

```
## S3 method for class 'grass_result'
plot(
  x,
  type = c("landing", "regime"),
  labels = c("auto", "inline", "legend"),
  show_medium = FALSE,
  title = NULL,
  subtitle = NULL,
  ...
)
```

Arguments

x	A grass_result object.
type	Either "landing" (default) or "regime".
labels	Where to identify the three curves: "auto" (default – inline at the right edge unless curves are too close together), "inline", or "legend".
show_medium	If TRUE, also plot the medium-quality reference curves as a faint dotted band. Default FALSE.
title, subtitle	Optional strings to override the default title and subtitle. Pass NULL to omit either.
...	Unused.

Details

Reference curves are the Youden-J-optimal metric values from the GRASS simulation across prevalence, for competent raters (Se, Sp at or above 0.85 for "high" quality; 0.70 for "medium").

Value

A ggplot object.

---

plot_surface *Standalone reference-surface plot for prospective study design*

Description

Renders the same context-conditioned reference surface that plot.grass_card(type = "surface") shows, but driven by scalar inputs rather than a fitted grass_report() result. The intended use case is prospective study design: a practitioner planning a multi-rater study at design (pi_hat, k, N) can ask "what does my reference surface look like for this metric, before I collect data?" – and plot_surface() answers without requiring a rating matrix.

Usage

```
plot_surface(
  metric,
  pi_hat = NULL,
  observed = NULL,
  k = NULL,
  N = NULL,
  axis = c("inter", "intra"),
  bands = c(0.5, 0.625, 0.75, 0.875, 1),
  ...
)
```

Arguments

metric	Character scalar. One of "pabak", "mean_pabak", "fleiss_kappa", "kappa", "mean_ac1", "ac1", "krippendorff_a". ICC is not supported (see Details).
pi_hat	Optional numeric scalar in (0, 1). The design's marginal positive rate. If supplied alone, draws a vertical reference line on the surface; if supplied together with observed, anchors the pin point.
observed	Optional numeric scalar. An observed (or hypothetical) coefficient value. When supplied with pi_hat, the function inverts to hat q and pins a marker at (pi_hat, hat q).
k	Optional integer. Rater count. Used for the subtitle only; non-ICC closed-form surfaces are k-invariant.
N	Optional integer. Sample size. Used for the subtitle only; non-ICC closed-form surfaces are N-invariant in the large-N limit the surface represents.
axis	One of "inter" (default) or "intra". Used for the subtitle only.
bands	Numeric length-5 vector giving band boundaries on q. Default c(0.5, 0.625, 0.75, 0.875, 1.0).
...	Reserved for future extension.

Details

The plot is a heatmap of the closed-form expectation E[metric](M_1, q) over the surface, where M_1 is the marginal positive rate and q in [0.5, 1] is the diagonal rater operating quality. Dotted contours overlay the four-band partition (bands argument). When pi_hat is supplied alone, a dashed vertical line marks the design's marginal. When both pi_hat and observed are supplied, a filled marker pins the implied (M_1, hat q) point on the surface; hat q is recovered by closed-form inversion of observed against the reference curve at pi_hat.

This function uses the closed-form non-ICC surfaces only (PABAK, Fleiss kappa, mean-pairwise AC1, Krippendorff alpha). ICC surfaces depend on the full subject-prevalence distribution F and the GLMM-gap-corrected fitted reference; those are accessed via position_on_surface() with metric = "icc" and (if available) a fitted card via plot.grass_card(type = "surface").

Value

A ggplot object.

See Also

plot.grass_card() for the card-driven surface view that adds card-specific annotations (band, qualifier, primary-coefficient pin); position_on_surface() for the underlying inversion machinery.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  # Bare surface -- what does PABAK's reference look like?
  plot_surface("pabak")

  # Mark a design context (no observed value yet -- pre-data).
  plot_surface("fleiss_kappa", pi_hat = 0.30, k = 5, N = 200)

  # Pin a hypothetical observation on the surface.
  plot_surface("mean_ac1", pi_hat = 0.30, observed = 0.62,
               k = 5, N = 200)
}
```

---

position_on_surface *Position an observed agreement coefficient on its DGP-calibrated surface*

Description

position_on_surface() is the Target-2 reporting primitive for the merged GRASS binary-rater-reliability paper. Given an observed coefficient value and the study design (pi_hat, k, N), it inverts the coefficient to an internal q_hat (rater operating quality on the Se = Sp diagonal under the clustered latent-class DGP), computes where the observed value sits on its reference sampling distribution (the surface percentile), and assigns a four-band adjective plus a confidence qualifier derived from the sampling-probability mass over bands.

Usage

```
position_on_surface(
  obs_value = NULL,
  metric,
  pi_hat = NULL,
  k = NULL,
  N = NULL,
  method = c("empirical", "delta"),
  bands = c(0.5, 0.625, 0.75, 0.875, 1),
  band_labels = c("Poor", "Moderate", "Strong", "Excellent"),
  surface_data = NULL,
  ratings = NULL,
  reference_type = c("fitted", "oracle"),
  ...
)
```

Arguments

obs_value	Numeric scalar. The observed agreement coefficient. Optional when ratings is supplied (auto-derived via compute_observed(metric, Y)).
metric	Character scalar. One of "pabak", "fleiss_kappa", "mean_ac1", "krippendorff_a", "icc".
pi_hat	Numeric scalar in (0, 1). The panel-identified marginal positive rate. Optional when ratings is supplied (auto-derived via mean(Y)); otherwise estimate from the rating matrix via grass_prevalence() or directly from rater marginals.
k	Integer >= 2. Number of raters. Optional when ratings is supplied (auto-derived as ncol(Y)).
N	Integer >= 1. Number of subjects. Optional when ratings is supplied (auto-derived as nrow(Y)).
method	One of "empirical" (default; uses the bundled sim-derived empirical q_hat sampling distribution) or "delta" (closed-form normal approximation from delta-method SE).
bands	Numeric vector of length 5 giving band boundaries on q. Must be strictly increasing, start at 0.5, end at 1.0. Default c(0.5, 0.625, 0.75, 0.875, 1.0).
band_labels	Character vector of length 4, one label per band. Default c("Poor", "Moderate", "Strong", "Excellent").
surface_data	Optional. A list with one or more of the following components, used when method = "empirical": • per_rep: a matrix of per-rep metric values at the closest simulated (q, pi_hat, k, N) cell – this is the empirical sampling distribution at that design. Used for the percentile of obs_value. • q_grid_per_rep: a matrix of dimension n_q x n_rep giving the empirical sampling distribution of the metric at each q-grid point, which the function uses to integrate P(q in band_j) against the inferred posterior shape of q_hat. • q_grid: numeric vector of q values for the rows of q_grid_per_rep. When surface_data is NULL and method = "empirical", the function signals an error with guidance.
ratings	Optional. From v0.2.0 this is the **primary input for all metrics** (not just ICC): supplying an N x k rating matrix auto-derives obs_value, pi_hat, k, and N. Accepts an N x k integer matrix of 0/1 values (rows = subjects, cols = raters), a data.frame with k rater columns, or a length-2 list of equal-length 0/1 vectors (k = 2). For metric = "icc", supplying ratings (also accepted as a long data.frame with columns subject and rating) additionally enables a glmer fit for (mu, tau2) that pins down the correct F_key for ICC inversion; without ratings, ICC falls back to a nearest-M1 F_key lookup with a prominent caveat note (tau2 is unidentified from pi_hat alone). The glmer path requires lme4 (Suggests).
reference_type	For metric = "icc" only. One of "fitted" (default; GLMM-gap-corrected reference matching what practitioners compute via glmer) or "oracle" (closed-form sigma^2_subject / (sigma^2_subject + pi^2/3) with sigma^2_subject

known from F). Use "oracle" only if obs_value was computed via oracle variance decomposition (non-standard for applied work). For N beyond the fitted-reference sim range (currently N > 200), the function auto-falls-back to oracle with an explanatory note.

... Reserved for future extension.

Details

The function implements the reporting convention locked on 2026-04-22: the practitioner cites the observed coefficient, its surface percentile, the band adjective, and the confidence qualifier. q_hat survives as internal scaffolding (surface parameterization and delta-method SE carrier) rather than as the labeled output.

Value

A list of class grass_surface_position with fields:

- observed_value – echo of obs_value
- metric – echo of metric
- design – list(pi_hat, k, N)
- q_hat – inverted rater quality (internal scaffold)
- se_q_hat – delta-method SE of q_hat
- percentile – surface percentile of obs_value in [0, 1]
- band_probabilities – named numeric vector of P(q in band_j)
- modal_band – integer index 1..4 of the highest-probability band
- modal_band_label – character adjective at the modal band
- confidence – "decisive" / "moderate" / "weak"
- sampling_method – which method was used
- notes – character vector of caveats (e.g. nearest-neighbor gaps)

Label construction (Sec.0.6.5)

Four adjacent-exclusive bands partition the above-chance quality range q in [0.5, 1.0] on equal width: **Poor** [0.5, 0.625), **Moderate** [0.625, 0.75), **Strong** [0.75, 0.875), **Excellent** [0.875, 1.0]. The geometric partition carries no a-priori weight; users may supply custom cut points through bands and matching band_labels.

The confidence qualifier is derived from the modal band's sampling probability p_star:

- decisive – p_star >= 0.90
- moderate – p_star in [0.60, 0.90)
- weak – p_star < 0.60

Sampling-probability construction

Two methods are implemented.

Delta method (`method = "delta"`, default): assumes `q_hat` is approximately normal with mean `q_hat` and standard deviation `se_q_hat` obtained from the delta-method SE of the monotone metric inversion, and computes `P(q in band_j)` by `pnorm()` on the band boundaries. This is the summary-stats-only fallback; it does not require the full simulation outputs.

Empirical method (`method = "empirical"`, default): uses the 2,000-replication sampling distribution from `multirater_sim_v3` (the symmetric-rater reference sim backing Paper 2) to form the empirical sampling distribution of `q_hat_rep` at the nearest matched scenario cell (`metric, F_key nearest to pi_hat, k nea`). The full per-rep data is not bundled (~300 MB); instead the package ships a precomputed 13-point empirical-quantile summary of the per-cell q_hat_rep vector. Band probabilities P(q_hat_rep in band_j) are recovered by piecewise-linear interpolation on the empirical CDF. Callers who want to pass their own per-rep vector may supply `surface_data$per_rep` or `surface_data$q_grid_per_rep`; those paths continue to override the bundled lookup. When the design (`pi_hat, k, N`) falls outside the simulated grid, nearest-neighbour clamping is applied and flagged in `notes`.

Internal reference-surface arithmetic

Under the clustered latent-class DGP with symmetric raters (`Se = Sp = q`), the large-N closed forms for PABAK, Fleiss kappa, AC1, and Krippendorff's alpha depend on `q` and the marginal positive rate `pi_+` only. This function uses `pi_hat` as a plug-in for `pi_+` and inverts the observed value on a 501-point q-grid on `[0.5, 1]` (matching `paper2/code/12_q_inversion.R` resolution). ICC requires the full subject-prevalence distribution F; in the absence of `surface_data` containing an ICC lookup, ICC requests fall through to a warning-noted delta-method approximation using the caller-supplied `q_hat_override` / `se_q_hat_override` if present, or stop with a clear message.

Ratings-primary path

From v0.2.0, the preferred entry point is to hand the rating matrix directly: `position_on_surface(ratings = Y, metric = "pabak")`. When `ratings` is supplied, the function auto-derives `obs_value` (via `compute_observed(metric, Y)`), `pi_hat` (`mean(Y)`), `k` (`ncol(Y)`), and `N` (`nrow(Y)`); any of those four arguments still supplied by the caller wins. This collapses the audit-style scalar-input path used in v0.1.x to a single matrix argument while keeping the scalar path callable for reproducibility checks. `ratings` accepts an `N x k` integer matrix in `{0, 1}`, a data.frame with `k` rater columns, or a length-2 list of equal-length 0/1 vectors (`k = 2`). Round-trip equality with the scalar path is a tested invariant.

See Also

`check_asymmetry()` for the companion Column A tier (rater asymmetry model-safety).

Examples

```
# Ratings-primary path: just hand it the matrix.
set.seed(1)
Y <- matrix(rbinom(1000, 1, 0.3), nrow = 200, ncol = 5)
position_on_surface(ratings = Y, metric = "pabak")

# Equivalent scalar-input path (audit):
```

```
position_on_surface(
  obs_value = 2 * mean(Y[, 1] == Y[, 2]) - 1,  # PABAK on first pair
  metric = "pabak", pi_hat = mean(Y), k = ncol(Y), N = nrow(Y)
)

# Delta-method positioning -- no sim data needed, summary-stats only.
r <- position_on_surface(
  obs_value = 0.62,
  metric    = "pabak",
  pi_hat    = 0.42,
  k         = 5,
  N         = 50
)
r$modal_band_label
r$confidence

# Fleiss kappa at imbalanced prevalence.
position_on_surface(
  obs_value = 0.18,
  metric    = "fleiss_kappa",
  pi_hat    = 0.08,
  k         = 3,
  N         = 200
)
```

`reset_grass_warnings` *Reset grass's once-per-session message cache*

Description

`grass` emits coercion messages and keyword-fallthrough warnings at most once per R session, keyed by the column name. When looping `grass_report()` over many studies with consistently named columns, only the first study in the loop sees the message. Call `reset_grass_warnings()` to clear the cache so the next call re-evaluates.

Usage

```
reset_grass_warnings()
```

Value

`invisible(NULL)`.

Examples

```
reset_grass_warnings()
```

scale_grass_metric *Color scale for grass metrics*

Description

Maps kappa, PABAK, and AC1 to the paper palette.

Usage

```
scale_color_grass_metric(...)

scale_fill_grass_metric(...)
```

Arguments

...	Passed through to ggplot2::scale_color_manual / ggplot2::scale_fill_manual.

Value

A ggplot2 scale.

theme_grass *ggplot2 theme for grass plots*

Description

Minimal publication-oriented theme. No chrome beyond what the data require.

Usage

```
theme_grass(base_size = 12)
```

Arguments

base_size	Base font size passed to ggplot2::theme_minimal().

Value

A ggplot2 theme object.

Index