Package 'modelskill'

September 17, 2026

Type Package

Title Assessing and Visualising the Performance of Prediction Models

Version 0.1.1

Description Provides tools for evaluating continuous predictions and their associated predictive uncertainty from statistical, machine-learning, geostatistical, and process-based models. It implements complementary measures of prediction error, association, agreement, efficiency, uncertainty calibration, and predictive-distribution performance, together with Taylor, solar, target, coverage, probability integral transform, and quantile-coverage diagnostics. Methods include the integrated evaluation approach of Wadoux, Walvoort and Brus (2022) <doi:10.1016/j.geoderma.2021.115332> and the uncertainty-validation framework of Schmidinger and Heuvelink (2023) <doi:10.1016/j.geoderma.2023.116585>.

License MIT + file LICENSE

URL https://github.com/AlexandreWadoux/modelskill, https://alexandrewadoux.github.io/modelskill/

BugReports https://github.com/AlexandreWadoux/modelskill/issues

Encoding UTF-8

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

Imports ggplot2 (>= 3.5.0), ggrepel, viridis

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

VignetteBuilder knitr

NeedsCompilation no

Author Alexandre M.J.-C. Wadoux [aut, cre, cph] (ORCID: <https://orcid.org/0000-0001-7325-9716>)

Maintainer Alexandre M.J.-C. Wadoux <alexandre.wadoux@yahoo.fr>

Repository CRAN

Date/Publication 2026-09-17 11:00:02 UTC

Contents

`accuracy_plot_metrics` *Summarize calibration from an accuracy plot*

Description

Computes numerical summaries of the departures from the 1:1 line in a prediction-interval reliability plot, also known in geostatistics as an accuracy plot. The approach evaluates prediction interval coverage probability (PICP) over multiple nominal interval levels.

Usage

```
accuracy_plot_metrics(
  obs,
  lower = NULL,
  upper = NULL,
  level = NULL,
  pred = NULL,
  predictive_sd = NULL,
  levels = NULL,
  na.rm = TRUE,
  distribution = NULL
)
```

Arguments

`obs`	Numeric observation vector.
`lower`, `upper`	Named lists of lower and upper prediction-interval bounds. Names must represent nominal coverage levels such as `"0.50"` or `"0.95"`. A single pair of numeric vectors is also accepted when `level` is supplied.
`level`	Nominal central prediction-interval coverage for a single numeric `lower`/`upper` pair. It must be `NULL` when named lists are supplied.
`pred`, `predictive_sd`	Optional numeric vectors of predictive means and predictive standard deviations. When supplied together, central prediction intervals are generated assuming normal predictive distributions. Do not also supply `lower` or `upper`.
`levels`	Nominal central prediction-interval coverage probabilities used when `pred` and `predictive_sd`, or `distribution`, are supplied. Values must lie strictly between zero and one. Defaults to every percentage from 1% to 99%.

`na.rm` Logical; remove incomplete observation/interval combinations? With interval lists, only cases complete in `obs` and both bounds at every supplied level are used, so all levels share the same validation sample. If `FALSE`, any incomplete case makes coverage missing at every level. With predictive samples, a row missing any draw or `obs` is incomplete.

`distribution` Optional numeric matrix or data frame of equally weighted predictive samples: one row per observation and one column per predictive draw. Supply this instead of bounds or predictive means and standard deviations. At least one draw is required; infinite values are not allowed.

Details

For a well-calibrated uncertainty model, the empirical coverage $\mathrm{PICP}(p)$ should be close to the nominal coverage probability p over the range of evaluated prediction intervals. Perfect calibration therefore corresponds to the 1:1 line.

The overall departure from this line is summarized by the absolute area

$$A = \int_0^1 |\mathrm{PICP}(p) - p| \, dp.$$

`absolute_deviation` is the exact area for the piecewise-linear interpolant of the evaluated coverage curve. Integration splits segments at crossings of the 1:1 line before applying the trapezoidal rule. A value of zero indicates perfect calibration; larger values indicate greater overall disagreement between nominal and empirical coverage.

The total deviation is further separated according to whether the empirical coverage curve lies above or below the 1:1 line:

$$A_{\mathrm{over}} = \int_0^1 \max\{\mathrm{PICP}(p) - p, 0\} \, dp,$$

$$A_{\mathrm{under}} = \int_0^1 \max\{p - \mathrm{PICP}(p), 0\} \, dp.$$

`over_uncertainty` is the area above the 1:1 line, where empirical coverage exceeds nominal coverage. This corresponds to over-coverage and is generally associated with prediction intervals that are too wide, or predictive uncertainty that is overestimated.

`under_uncertainty` is the area below the 1:1 line, where empirical coverage is smaller than nominal coverage. This corresponds to under-coverage and is generally associated with prediction intervals that are too narrow, or predictive uncertainty that is underestimated.

`over_percent` and `under_percent` give the relative contributions of these two components to the total absolute deviation. They describe percentages of the total area of miscalibration, not percentages of observations. When `absolute_deviation` is greater than zero, the two percentages sum to 100.

The integration is performed over the supplied nominal levels after adding the endpoints `(0, 0)` and `(1, 1)` to the reliability curve. For a normal predictive distribution, supplying `pred` and `predictive_sd` without `levels` evaluates the default sequence of nominal interval levels from 1% to 99%. The added endpoints are assumptions, not measured coverage values; the resulting

areas depend on the supplied grid and this interpolation. Interval lists use cases complete across all levels, as in `gg_coverage()`. If coverage cannot be calculated, all five summaries are `NA` with a warning. When total deviation is zero, the two percentage contributions are `NA` because there is no deviation to apportion; this does not issue a warning.

These summaries describe calibration rather than sharpness. They should therefore be interpreted together with measures such as prediction interval width or a proper scoring rule when comparing predictive uncertainty.

Accuracy plots were proposed for the direct assessment of local uncertainty by Deutsch (1997) and subsequently applied to geostatistical uncertainty evaluation in soil science by Goovaerts (2001). Related numerical summaries of departure from the accuracy-plot reference line were used by Wadoux, Brus, and Heuvelink (2018).

Predictive samples can be supplied directly using `distribution`. Central empirical intervals are generated at `levels` as in `gg_coverage()`, using type-7 quantiles and the same complete cases at every level.

Value

A one-row data frame containing:

absolute_deviation Total area between the empirical coverage curve and the 1:1 line. Zero is ideal.

over_uncertainty Area above the 1:1 line, corresponding to over-coverage.

under_uncertainty Area below the 1:1 line, corresponding to under-coverage.

over_percent Percentage of total absolute deviation occurring above the 1:1 line.

under_percent Percentage of total absolute deviation occurring below the 1:1 line.

References

Deutsch, C. V. (1997). Direct assessment of local accuracy and precision. In E. Y. Baafi and N. A. Schofield (Eds.), *Geostatistics Wollongong '96*, pp. 115-125.

Goovaerts, P. (2001). Geostatistical modelling of uncertainty in soil science. *Geoderma*, 103, 3-26. doi:10.1016/S0016-7061(01)00067-2

Wadoux, A. M. J.-C., Brus, D. J. and Heuvelink, G. B. M. (2018). Accounting for non-stationary variance in geostatistical mapping of soil properties. *Geoderma*, 324, 138-147.

See Also

`gg_coverage()`, `picp()`, `coverage_error()`, `interval_width()`, `interval_score()`

Examples

```
set.seed(123)
n <- 200
pred <- seq(0, 10, length.out = n)
predictive_sd <- rep(1, n)
obs <- stats::rnorm(n, mean = pred, sd = predictive_sd)

accuracy_plot_metrics(
```

```
  obs,
  pred = pred,
  predictive_sd = predictive_sd
)
accuracy_plot_metrics(1:3, distribution = cbind(0:2, 1:3, 2:4),
                      levels = c(0.5, 0.9))
```

`bias` *Mean error (ME) of quantitative predictions*

Description

Mean error (ME; also called bias) is the mean signed difference between observations and predictions, calculated as observation minus prediction.

Usage

```
bias(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\mathrm{ME} = \frac{1}{n}\sum_{i=1}^{n}(obs_i - pred_i)$$

An ME of zero indicates no average systematic error. Negative values indicate overprediction on average, whereas positive values indicate underprediction on average. ME has the same units as the response variable. Opposing errors can cancel, so interpret ME together with an unsigned error measure such as `mae()` or `rmse()`. Missing pairs are removed when `na.rm = TRUE`; otherwise the result is `NA` when any pair is missing.

Value

One numeric value.

References

Legates, D. R. and McCabe, G. J. (1999). Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation. *Water Resources Research*, 35(1), 233-241. doi: 10.1029/1998WR900018

Examples

```
bias(c(1, 2, 3), c(1, 3, 2))
```

`ccc` *Lin's concordance correlation coefficient*

Description

Lin's concordance correlation coefficient (CCC; ρ_c) measures agreement between observations and predictions by combining Pearson correlation with differences in location and scale. Unlike Pearson correlation alone, CCC evaluates how closely paired values approach the line of equality.

Usage

```
ccc(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\rho_c = \frac{2\sum_{i=1}^{n}(obs_i - \bar{obs})(pred_i - \bar{pred})}{\sum_{i=1}^{n}(obs_i - \bar{obs})^2 + \sum_{i=1}^{n}(pred_i - \bar{pred})^2 + n(\bar{obs} - \bar{pred})^2}.$$

CCC ranges from -1 to 1, with 1 indicating perfect agreement. Values decrease as observations and predictions differ in linear association, mean, or scale. Negative values indicate negative concordance.

CCC can also be expressed as

$$\rho_c = rC_b,$$

where r is the Pearson correlation coefficient and C_b is a bias-correction factor that accounts for departures from the line of equality. Consequently, CCC incorporates both association and agreement into a single statistic.

When CCC is used to evaluate predictive models, its value is most informative when considered together with complementary statistics. Different combinations of correlation, mean bias, and scale differences can produce similar CCC values, so the coefficient alone does not identify which component is responsible for disagreement between observations and predictions. In addition, CCC depends partly on the variability of the reference observations. Direct comparison of CCC values obtained from substantially different datasets or target populations should therefore be made with caution.

For prediction-model evaluation, CCC can usefully be reported alongside measures describing individual aspects of predictive performance, such as `bias()`, `mae()`, `rmse()`, `correlation()`, or `R2()`. This allows the overall concordance indicated by CCC to be interpreted together with the magnitude and sources of prediction error.

Population variances (divisor n) are used, matching the package convention. If either vector is constant and the displayed denominator is positive, CCC is zero, including a single unequal observation-prediction pair. Identical constant vectors have a zero denominator and return `NA` with a warning, as do inputs with no valid pairs. These conventions are symmetric in observations and predictions. Missing-value handling follows `bias()`.

Value

One numeric value between -1 and 1, with 1 indicating perfect agreement.

References

Lin, L. I.-K. (1989). A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, 45, 255-268. doi:10.2307/2532051

Wadoux, A. M. J.-C. and Minasny, B. (2024). Some limitations of the concordance correlation coefficient to characterise model accuracy. *Ecological Informatics*, 83, 102820. doi:10.1016/j.ecoinf.2024.102820

See Also

`correlation()`, `bias()`, `mae()`, `rmse()`, `R2()`

Examples

```
obs <- c(1, 2, 3, 4, 5)

# Perfect agreement
ccc(obs, obs)

# Systematic bias reduces concordance
ccc(obs, obs + 1)

# Compare with Pearson correlation
correlation(obs, obs + 1)
ccc(obs, obs + 1)
```

`correlation` *Pearson correlation*

Description

Pearson product-moment correlation between observations and predictions.

Usage

```
correlation(obs, pred, na.rm = TRUE)
```

Arguments

obs	Numeric observation vector.
pred	Numeric prediction vector paired with obs.
na.rm	Logical; remove incomplete pairs?

Details

$$r = \frac{\sum_{i=1}^{n}(obs_i - \bar{obs})(pred_i - \bar{pred})}{\sqrt{\sum_{i=1}^{n}(obs_i - \bar{obs})^2 \sum_{i=1}^{n}(pred_i - \bar{pred})^2}}$$

Correlation ranges from -1 to 1: one indicates a perfect increasing linear association, minus one a perfect decreasing linear association, and zero no linear association. It returns NA with a warning when fewer than two valid pairs remain or either vector has zero variance. Correlation is unaffected by additive bias and proportional scaling, so it measures pattern association rather than agreement or prediction accuracy. Interpret it with bias(), rmse(), and an agreement measure such as ccc().

Value

One numeric value.

References

Willmott, C. J. (1984). On the evaluation of model performance in physical geography. In G. L. Gaile and C. J. Willmott (Eds.), *Spatial Statistics and Models* (pp. 443-460). D. Reidel.

Legates, D. R. and McCabe, G. J. (1999). Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation. *Water Resources Research*, 35(1), 233-241. doi:10.1029/1998WR900018

Examples

```
correlation(1:3, c(1, 3, 2))
```

---

coverage *Empirical prediction-interval coverage*

Description

Backward-compatible alias for picp(). New code should prefer picp(), the conventional abbreviation for prediction interval coverage probability. Its equation, interpretation, and reference are given in picp().

Usage

```
coverage(
  obs,
  lower = NULL,
  upper = NULL,
  na.rm = TRUE,
  level = 0.95,
  pred = NULL,
  predictive_sd = NULL,
  distribution = NULL
)
```

Arguments

`obs`	Numeric observation vector.
`lower`, `upper`	Optional numeric lower and upper prediction-interval bounds. Supply both, without another input representation.
`na.rm`	Logical; remove incomplete cases? See Input representations.
`level`	Nominal central interval coverage, strictly between zero and one. Used to generate bounds from predictive means and standard deviations or predictive samples; it does not alter explicit bounds.
`pred`, `predictive_sd`	Optional numeric vectors of predictive means and predictive standard deviations, supplied together and of the same length as `obs`. Assumes normal predictive distributions. Non-missing predictive standard deviations must be finite and strictly positive.
`distribution`	Optional numeric matrix or data frame of equally weighted predictive samples: one row per observation and one column per predictive draw. Supply this instead of bounds or predictive means and standard deviations. At least one draw is required; infinite values are not allowed.

Value

One numeric value on the probability scale from zero to one.

Input representations

Supply exactly one of explicit `lower` and `upper` bounds, predictive mean and standard deviation (`pred` and `predictive_sd`), or predictive samples (`distribution`). Normal inputs generate central intervals using normal quantiles. Samples generate equal-tailed intervals using `stats::quantile()` with `type = 7`, at probabilities `(1 - level) / 2` and `(1 + level) / 2`. With `na.rm = TRUE`, a case is removed if its observation or any supplied predictive value is missing; individual missing draws are not discarded within a case. With `na.rm = FALSE`, incomplete inputs give `NA`. No complete cases also gives `NA`. Standalone `interval_width()` ignores missing `obs`.

Examples

```
coverage(1:3, c(0, 1, 2), c(2, 3, 4))
```

`coverage_error` *Prediction-interval coverage error*

Description

Empirical `picp()` minus nominal coverage `level`. Positive values mean over-coverage and negative values mean under-coverage. The result is on the probability scale; multiply by 100 for percentage points.

Usage

```
coverage_error(
  obs,
  lower = NULL,
  upper = NULL,
  level = 0.95,
  na.rm = TRUE,
  pred = NULL,
  predictive_sd = NULL,
  distribution = NULL
)
```

Arguments

`obs`	Numeric observation vector.
`lower`, `upper`	Optional numeric lower and upper prediction-interval bounds. Supply both, without another input representation.
`level`	Nominal central interval coverage, strictly between zero and one.
`na.rm`	Logical; remove incomplete cases? See Input representations.
`pred`, `predictive_sd`	Optional numeric vectors of predictive means and predictive standard deviations, supplied together and of the same length as `obs`. Assumes normal predictive distributions. Non-missing predictive standard deviations must be finite and strictly positive.
`distribution`	Optional numeric matrix or data frame of equally weighted predictive samples: one row per observation and one column per predictive draw. Supply this instead of bounds or predictive means and standard deviations. At least one draw is required; infinite values are not allowed.

Details

$$\text{PICP error}(\tau) = \text{PICP}(\tau) - \tau$$

Zero is ideal. Positive values mean intervals cover too often (are too wide or over-pessimistic); negative values mean intervals cover too rarely.

Value

One numeric value.

Input representations

Supply exactly one of explicit `lower` and `upper` bounds, predictive mean and standard deviation (`pred` and `predictive_sd`), or predictive samples (`distribution`). Normal inputs generate central intervals using normal quantiles. Samples generate equal-tailed intervals using `stats::quantile()` with `type = 7`, at probabilities `(1 - level) / 2` and `(1 + level) / 2`. With `na.rm = TRUE`, a case is removed if its observation or any supplied predictive value is missing; individual missing draws are not discarded within a case. With `na.rm = FALSE`, incomplete inputs give `NA`. No complete cases also gives `NA`. Standalone `interval_width()` ignores missing `obs`.

References

Schmidinger, J. and Heuvelink, G. B. M. (2023). Validation of uncertainty predictions in digital soil mapping. *Geoderma*, 437, 116585. doi:10.1016/j.geoderma.2023.116585

Examples

```
coverage_error(1:3, c(0, 1, 2), c(2, 3, 4), level = .8)
```

`crmse` *Centered root mean squared error*

Description

Centered RMSE (cRMSE) is the root mean square difference after removing the mean error from the paired errors.

Usage

```
crmse(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\mathrm{cRMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[(obs_i - pred_i) - \frac{1}{n}\sum_{j=1}^{n}(obs_j - pred_j)\right]^2}.$$

cRMSE is non-negative, has the response units, and zero is ideal. It measures disagreement in pattern and spread independently of a constant mean bias. Interpret it with `bias()` because two models with identical cRMSE can have different systematic errors. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Taylor, K. E. (2001). Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research*, 106, 7183-7192. doi:10.1029/2000JD900719

Examples

```
crmse(1:3, c(1, 3, 2))
```

---

crps *Continuous ranked probability score*

Description

CRPS compares a predictive distribution with an observation; lower values are better. Supply either equally weighted predictive samples in `distribution`, or a normal predictive distribution through `pred` and `predictive_sd`.

Usage

```
crps(obs, distribution = NULL, pred = NULL, predictive_sd = NULL, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`distribution`	Numeric matrix/data frame of equally weighted predictive samples, one row per observation.
`pred`, `predictive_sd`	Mean and strictly positive predictive SD for normal predictive distributions.
`na.rm`	Logical; remove incomplete observation/distribution rows?

Details

$$\mathrm{CRPS}(F, obs) = \int_{-\infty}^{\infty} [F(z) - I(z \geq obs)]^2 \, dz$$

CRPS has response units and lower values are better; zero is ideal. It is a proper scoring rule that jointly rewards calibrated and sharp distributions.

Value

One numeric mean CRPS value.

References

Hersbach, H. (2000). Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, 15, 559-570. doi:10.1175/1520-0434(2000)015%3C0559:DOTCRP%3E2.0.CO;2

Examples

```
crps(0, distribution = matrix(c(-1, 1), nrow = 1))
```

`crps_decomposition` *CRPS reliability decomposition*

Description

Decomposes mean ensemble CRPS into a reliability component and potential CRPS following Hersbach (2000). Predictive-distribution columns are treated as equally likely ensemble members. This function accepts predictive samples only, not a predictive mean and standard deviation. Its decomposition is defined for finite ensembles.

Usage

```
crps_decomposition(obs, distribution, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`distribution`	Numeric matrix/data frame of equally weighted predictive samples, one row per observation.
`na.rm`	Logical; remove incomplete observation/distribution rows?

Details

$$\text{CRPS} = \text{RELI} + \text{potential CRPS}$$

`reliability` (RELI) is non-negative and zero is ideal; smaller values indicate better distributional calibration. `potential_crps` is the remainder after removing reliability error. The decomposition is applicable here only to equally weighted predictive samples.

For each retained case $i = 1, \ldots, n$, let $x_{i,1} \le \cdots \le x_{i,m}$ be the sorted ensemble members and $p_j = j/m$. For interior bins $j = 1, \ldots, m-1$, define

$$\alpha_{i,j} = \max\{0, \min(obs_i, x_{i,j+1}) - x_{i,j}\}, \qquad \beta_{i,j} = \max\{0, x_{i,j+1} - \max(obs_i, x_{i,j})\}.$$

These are the portions of a bin below and above the observation. Equality at a bin edge assigns the full width to the appropriate portion; tied ensemble members have zero width. With bars denoting means over cases,

$$g_j = \bar{\alpha}_j + \bar{\beta}_j, \qquad o_j = \frac{\bar{\beta}_j}{g_j}.$$

If $g_j = 0$, set $o_j = p_j$; this bin contributes zero. The two exterior bins use

$$o_0 = \frac{1}{n}\sum_{i=1}^{n} I(obs_i \le x_{i,1}), \qquad o_m = \frac{1}{n}\sum_{i=1}^{n} I(obs_i \le x_{i,m}),$$

$$g_0 = \frac{\frac{1}{n}\sum_{i=1}^{n} \max(x_{i,1} - obs_i, 0)}{o_0}, \qquad g_m = \frac{\frac{1}{n}\sum_{i=1}^{n} \max(obs_i - x_{i,m}, 0)}{1 - o_m}.$$

Set $g_0 = 0$ when $o_0 = 0$ and $g_m = 0$ when $o_m = 1$. The components, in response units, are then

$$\text{RELI} = \sum_{j=0}^{m} g_j (o_j - p_j)^2, \qquad \text{potential CRPS} = \sum_{j=0}^{m} g_j o_j (1 - o_j).$$

Value

One-row data frame with `crps`, `reliability`, and `potential_crps`.

References

Hersbach, H. (2000). Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, 15, 559-570. doi:10.1175/1520-0434(2000)015%3C0559:DOTCRP%3E2.0.CO;2

Examples

```
crps_decomposition(c(0, 1), matrix(c(-1, 1, 0, 2), nrow = 2))
```

`diagram_stats` *Statistics underlying summary diagrams*

Description

Computes the statistics used as coordinates in Taylor, solar, and target diagrams without constructing a plot. This is useful for inspecting the numerical quantities represented by the diagrams or for constructing custom visualizations.

Usage

```
diagram_stats(mods, obs, na.rm = TRUE)
```

Arguments

`mods`	Numeric vector, list of numeric vectors, or numeric matrix/data frame with one model per column. Rows match `obs` in order. Supplied model names must be unique; missing names are generated.
`obs`	A numeric observation vector.
`na.rm`	Logical; remove incomplete observation-prediction pairs separately for each model? If `FALSE`, missing pairs cause an error because diagram coordinates cannot be calculated. The default is `TRUE`.

Details

Inputs are paired by position. At least two complete pairs and non-zero observation standard deviation are required for every model.

For the diagram geometry, standard deviations are calculated as population moments, using divisor n, rather than the $n - 1$ sample standard deviation returned by `stats::sd()`. This convention makes the normalized error decomposition exact for finite samples.

Let

$$e_i = obs_i - pred_i$$

denote the prediction error, and let σ_p and σ_o denote the population-moment standard deviations of predictions and observations, respectively.

The standard-deviation ratio is

$$\sigma^* = \frac{\sigma_p}{\sigma_o}.$$

The normalized mean error is

$$\mathrm{ME}^* = \frac{\bar{e}}{\sigma_o}.$$

Positive ME* indicates underprediction and negative ME* indicates overprediction under the package convention `obs - pred`.

Let σ_e denote the population-moment standard deviation of the errors. The standardized error standard deviation is

$$\mathrm{SDE}^* = \frac{\sigma_e}{\sigma_o}.$$

Using the relationship between the variances of observations, predictions, and their errors, this is equivalently

$$\mathrm{SDE}^* = \sqrt{1 + \sigma^{*2} - 2\sigma^* r}.$$

The sign used for `signed_sde` indicates whether the prediction standard deviation is smaller or larger than the observation standard deviation: negative when $\sigma_p < \sigma_o$ and positive when $\sigma_p \geq \sigma_o$. Equal standard deviations therefore receive a positive sign, following the original diagram implementation.

Constant predictions have undefined Pearson correlation and are returned with Pearson correlation set to `NA`, an SD ratio of zero, and SDE equal to one. Their diagram geometry remains defined even though their correlation is not.

With this common population-moment normalization, the exact finite-sample relationship is

$$\frac{\mathrm{RMSE}^2}{\sigma_o^2} = \mathrm{ME}^{*2} + \mathrm{SDE}^{*2}.$$

Thus Euclidean distance from the origin in the solar and target diagrams is exactly RMSE normalized by the population-moment observation standard deviation.

Value

A data frame with one row per model and the following columns:

model Model identifier.

n Number of complete observation-prediction pairs.

r Pearson correlation coefficient.

sd_ratio Ratio of prediction to observation standard deviation.

mean_error Mean error, calculated as observation minus prediction, in the original response units.

nME Normalized mean error (ME*), obtained by dividing mean error by the population-moment standard deviation of the observations.

sde Standardized standard deviation of the error (SDE*), obtained by dividing the population-moment standard deviation of the errors by the population-moment standard deviation of the observations.

signed_sde SDE* multiplied by the sign of the difference between prediction and observation standard deviations.

References

Wadoux, A. M. J.-C., Walvoort, D. J. J., and Brus, D. J. (2022). An integrated approach for the evaluation of quantitative soil maps through Taylor and solar diagrams. *Geoderma*, 405, 115332. doi:10.1016/j.geoderma.2021.115332

See Also

model_metrics(), gg_taylor(), gg_solar(), gg_target()

Examples

```
obs <- c(1, 2, 3, 4, 5)

mods <- list(
  perfect = obs,
  biased = obs + 1,
  noisy = c(1, 3, 2, 5, 4)
)

diagram_stats(mods, obs)
```

---

gg_coverage *Plot prediction-interval calibration*

Description

Produces a prediction interval coverage probability (PICP) reliability plot, also known in geostatistics as an accuracy plot. The plot compares empirical prediction-interval coverage with the corresponding nominal coverage over one or more central prediction-interval levels.

Usage

```
gg_coverage(
  obs,
  lower = NULL,
  upper = NULL,
  level = NULL,
  pred = NULL,
  predictive_sd = NULL,
  levels = NULL,
  na.rm = TRUE,
  point_size = 3,
  line_width = 0.6,
  distribution = NULL
)
```

Arguments

obs	Numeric observation vector.
lower, upper	Named lists of lower and upper prediction-interval bounds. Names must represent nominal coverage levels such as "0.50" or "0.95". A single pair of numeric vectors is also accepted when level is supplied.
level	Nominal central prediction-interval coverage for a single numeric lower/upper pair. It must be NULL when named lists are supplied.
pred, predictive_sd	Optional numeric vectors of predictive means and predictive standard deviations. When supplied together, central prediction intervals are generated assuming normal predictive distributions. Do not also supply lower or upper.
levels	Nominal central prediction-interval coverage probabilities used when pred and predictive_sd, or distribution, are supplied. Values must lie strictly between zero and one. Defaults to every percentage from 1% to 99%.
na.rm	Logical; remove incomplete observation/interval combinations? With interval lists, only cases complete in obs and both bounds at every supplied level are used, so all levels share the same validation sample. If FALSE, any incomplete case makes coverage missing at every level. With predictive samples, a row missing any draw or obs is incomplete.
point_size	Positive numeric point size.
line_width	Positive numeric width of the 1:1 reference line.
distribution	Optional numeric matrix or data frame of equally weighted predictive samples: one row per observation and one column per predictive draw. Supply this instead of bounds or predictive means and standard deviations. At least one draw is required; infinite values are not allowed.

Details

For a nominal central prediction interval with coverage probability p, a calibrated predictive uncertainty model should contain approximately a proportion p of independent validation observations. Consequently, empirical PICP should satisfy

$$\mathrm{PICP}(p) \approx p,$$

and a well-calibrated model should follow the dashed 1:1 reference line.

Points below the 1:1 line indicate **under-coverage**: fewer observations are contained in the prediction intervals than expected. This generally indicates prediction intervals that are too narrow and predictive uncertainty that is underestimated.

Points above the 1:1 line indicate **over-coverage**: more observations are contained in the prediction intervals than expected. This generally indicates prediction intervals that are wider than required and predictive uncertainty that is overestimated.

Evaluating several interval levels provides more information than evaluating a single PICP value because it shows how calibration changes across the predictive distribution. When pred and predictive_sd are supplied and levels is left NULL, central normal prediction intervals are evaluated from 1% to 99% nominal coverage. Alternatively, distribution generates central empirical intervals from

equally weighted predictive samples, using `stats::quantile()` with `type = 7`. The same default levels apply. Supply only one input representation.

The accuracy-plot approach was developed for direct assessment of local uncertainty in geostatistics by Deutsch (1997) and subsequently applied to uncertainty evaluation in soil science by Goovaerts (2001). Similar reliability diagnostics have also been used in digital soil mapping, including Wadoux, Brus, and Heuvelink (2018).

The graphical departures from the 1:1 line can be summarized numerically with `accuracy_plot_metrics()`, which calculates the total absolute area between the empirical coverage curve and the reference line and separates this departure into over-coverage and under-coverage components.

Central PICP evaluates the **joint coverage** of lower and upper prediction-interval bounds. It therefore does not identify how non-coverage is distributed between the two tails. A model can have approximately correct central interval coverage while having too many observations below one bound and too few above the other. Use `gg_qcp()` or `gg_pit()` when tail-specific or distributional calibration is also of interest.

Calibration should also be distinguished from sharpness. Good coverage can be obtained using unnecessarily wide prediction intervals, so PICP calibration should generally be interpreted together with interval width or a proper scoring rule such as `interval_score()`.

Value

A `ggplot2` object. Its plotting data contain:

nominal Nominal prediction-interval coverage probability.

picp Empirical prediction interval coverage probability.

References

Deutsch, C. V. (1997). Direct assessment of local accuracy and precision. In E. Y. Baafi and N. A. Schofield (Eds.), *Geostatistics Wollongong '96*, pp. 115-125.

Goovaerts, P. (2001). Geostatistical modelling of uncertainty in soil science. *Geoderma*, 103, 3-26. doi:10.1016/S0016-7061(01)00067-2

Wadoux, A. M. J.-C., Brus, D. J. and Heuvelink, G. B. M. (2018). Accounting for non-stationary variance in geostatistical mapping of soil properties. *Geoderma*, 324, 138-147.

Schmidinger, J. and Heuvelink, G. B. M. (2023). Validation of uncertainty predictions in digital soil mapping. *Geoderma*, 437, 116585. doi:10.1016/j.geoderma.2023.116585

See Also

`picp()`, `coverage_error()`, `accuracy_plot_metrics()`, `gg_qcp()`, `gg_pit()`, `interval_width()`, `interval_score()`

Examples

```
set.seed(123)

n <- 200
pred <- seq(0, 10, length.out = n)
predictive_sd <- rep(1, n)
```

```
obs <- stats::rnorm(n, mean = pred, sd = predictive_sd)

# Reliability curve generated directly from a normal predictive distribution
p_normal <- gg_coverage(
  obs,
  pred = pred,
  predictive_sd = predictive_sd
)

# Selected prediction intervals can also be supplied directly
lower <- list(
  `0.50` = pred + stats::qnorm(0.25) * predictive_sd,
  `0.90` = pred + stats::qnorm(0.05) * predictive_sd
)

upper <- list(
  `0.50` = pred + stats::qnorm(0.75) * predictive_sd,
  `0.90` = pred + stats::qnorm(0.95) * predictive_sd
)

p_intervals <- gg_coverage(obs, lower = lower, upper = upper)
p_samples <- gg_coverage(1:3, distribution = cbind(0:2, 1:3, 2:4),
                         levels = c(0.5, 0.9))
# Print p_normal, p_intervals or p_samples to display a plot.
```

`gg_pit` *Plot a probability integral transform histogram*

Description

Produces a probability integral transform (PIT) histogram for assessing the calibration of continuous predictive distributions.

Usage

```
gg_pit(pit_values, bins = 10, na.rm = TRUE)
```

Arguments

`pit_values`	Numeric vector of PIT values between zero and one, typically calculated with `pit()`.
`bins`	Positive integer number of equal-width histogram bins.
`na.rm`	Logical; remove missing PIT values? The default is `TRUE`.

Details

For observation y_i with predictive cumulative distribution function F_i, the PIT value is

$$u_i = F_i(y_i).$$

If the predictive distributions are calibrated and continuous, the PIT values should be approximately uniformly distributed between zero and one. The dashed horizontal line shows the density expected under a uniform distribution.

Departures from uniformity can indicate systematic miscalibration. Common patterns include:

- a U-shaped histogram, with excess values near zero and one, which is commonly associated with predictive distributions that are too narrow (underdispersed);
- a hump-shaped histogram, with excess values near 0.5, which is commonly associated with predictive distributions that are too wide (overdispersed);
- an excess of PIT values near zero, which can occur when predictions are systematically too high relative to the observations;
- an excess of PIT values near one, which can occur when predictions are systematically too low relative to the observations.

These patterns are diagnostic rather than unique: different forms of misspecification can produce similar PIT histograms. The PIT should therefore be interpreted together with other calibration and performance diagnostics.

PIT histograms assess the calibration of the complete predictive distribution. This differs from `gg_coverage()`, which evaluates central prediction-interval coverage, and `gg_qcp()`, which evaluates calibration of individual predictive quantiles.

The number of histogram bins affects the appearance of the diagnostic. Too few bins may conceal departures from uniformity, whereas too many bins can make sampling variability appear as structure, particularly for small validation datasets.

Value

A `ggplot2` object showing the empirical PIT density. Under uniform calibration the expected density is one.

References

Gneiting, T., Balabdaoui, F. and Raftery, A. E. (2007). Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B*, 69, 243-268. doi:10.1111/j.1467-9868.2007.00587.x

Schmidinger, J. and Heuvelink, G. B. M. (2023). Validation of uncertainty predictions in digital soil mapping. *Geoderma*, 437, 116585. doi:10.1016/j.geoderma.2023.116585

See Also

`pit()`, `gg_qcp()`, `gg_coverage()`

Examples

```
set.seed(123)

# Approximately calibrated PIT values
values <- stats::runif(500)

gg_pit(values)

# Use more bins
gg_pit(values, bins = 20)
```

---

gg_qcp *Quantile calibration plot*

Description

Plots empirical quantile coverage probability (QCP) against the corresponding nominal predictive quantile levels. For a calibrated predictive distribution, approximately a proportion p of observations should fall below the predicted p-quantile. Points should therefore follow the dashed 1:1 line.

Usage

```
gg_qcp(
  obs,
  quantiles = NULL,
  levels = NULL,
  pred = NULL,
  predictive_sd = NULL,
  na.rm = TRUE,
  point_size = 3,
  distribution = NULL
)
```

Arguments

obs	Numeric observation vector.
quantiles	Numeric matrix or data frame containing predicted quantiles in columns. Required unless pred and predictive_sd, or distribution, are supplied.
levels	Numeric vector of nominal quantile probabilities corresponding to the columns of quantiles. With predictive means and standard deviations or predictive samples, defaults to seq(0.05, 0.95, by = 0.05).
pred	Optional numeric vector of predictive means. Must be supplied together with predictive_sd; in this mode predictive quantiles are generated assuming normal predictive distributions.
predictive_sd	Optional numeric vector of predictive standard deviations. Must be supplied together with pred.

na.rm	Logical; remove incomplete observation/quantile rows?
point_size	Positive numeric point size.
distribution	Optional numeric matrix or data frame of equally weighted predictive samples: one row per observation and one column per predictive draw. Supply this instead of bounds or predictive means and standard deviations. At least one draw is required; infinite values are not allowed.

Details

Unlike gg_coverage(), which evaluates the joint coverage of central prediction intervals, gg_qcp() evaluates individual predictive quantiles. It can therefore reveal asymmetric or one-sided miscalibration that may be hidden when lower and upper prediction-interval bounds are assessed together.

Quantiles can be supplied directly through quantiles and levels. When the predictive distribution is assumed to be normal, they can instead be generated automatically from predictive means (pred) and predictive standard deviations (predictive_sd). In this mode, the default is to evaluate quantiles from 0.05 to 0.95 in increments of 0.05. Predictive samples (distribution) are also accepted, using empirical quantiles (stats::quantile(), type = 7) at the same default levels. Supply exactly one representation. As in qcp(), every level uses the same complete rows across obs and all supplied predictive values. With na.rm = FALSE, any incomplete row makes coverage missing at every level.

A calibration curve is generally most informative when quantiles are available over a reasonably dense range of probability levels. A smaller number of levels remains valid when only selected predictive quantiles are available.

Value

A ggplot2 object with nominal and qcp data columns.

See Also

qcp(), gg_coverage(), gg_pit()

Examples

```
set.seed(123)

n <- 500
pred <- seq(0, 10, length.out = n)
predictive_sd <- rep(1, n)
obs <- stats::rnorm(n, mean = pred, sd = predictive_sd)

# Generate a quantile-calibration curve directly from a normal
# predictive distribution
gg_qcp(
  obs,
  pred = pred,
  predictive_sd = predictive_sd
)
```

```
# Predicted quantiles can also be supplied directly
levels <- seq(0.05, 0.95, by = 0.05)

quantiles <- vapply(
  levels,
  function(p) pred + stats::qnorm(p) * predictive_sd,
  numeric(n)
)

gg_qcp(
  obs,
  quantiles = quantiles,
  levels = levels
)
gg_qcp(1:3, distribution = cbind(0:2, 1:3, 2:4), levels = c(0.25, 0.75))
```

---

gg_solar *Create a solar diagram*

Description

Creates a solar diagram comparing quantitative prediction models with an observation vector. The horizontal coordinate is normalized mean error (ME*) and the vertical coordinate is standardized unbiased root mean square difference (SDE*).

Usage

```
gg_solar(
  mods,
  obs,
  colorval = NULL,
  colorval.name = NULL,
  colour_by = c("efficiency", "model", "correlation", "r2"),
  x.axis_begin = -1,
  x.axis_end = 1,
  y.axis_end = 1.1,
  by = 0.1,
  label = FALSE,
  point_size = 7,
  label_size = 4,
  na.rm = TRUE,
  legend = TRUE,
  reference = TRUE
)
```

Arguments

mods	Numeric vector, list of numeric vectors, or numeric matrix/data frame with one model per column. Rows match obs in order. Supplied model names must be unique; missing names are generated.
obs	A numeric observation vector.
colorval	Optional finite numeric vector with one value per model, in model order (names do not reorder values). When supplied, it overrides the continuous metric selected by colour_by.
colorval.name	Optional point-colour legend title. The title can also be replaced afterwards with labs(colour = ...).
colour_by	Character string defining point colouring. "efficiency" uses R-squared / NSE / MEC and is the default; "model" gives every model a categorical colour and model-name legend; "correlation" uses Pearson correlation; and "r2" uses squared Pearson correlation.
x.axis_begin	Lower endpoint of the manually drawn horizontal reference axis.
x.axis_end	Upper endpoint of the manually drawn horizontal reference axis.
y.axis_end	Upper endpoint of the manually drawn vertical reference axis. Defaults to 1.1 so that the outer correlation circle at 1 and points lying on it remain clearly visible.
by	Spacing between manually drawn reference-axis ticks.
label	Logical; draw model names beside points using ggrepel?
point_size	Numeric point size.
label_size	Numeric size of model labels.
na.rm	Logical; remove incomplete observation-prediction pairs separately for each model? If FALSE, missing pairs cause an error because diagram coordinates cannot be calculated. The default is TRUE.
legend	Logical; show the point-colour and correlation-region legends?
reference	Logical; draw the original correlation regions and outer reference circle?

Details

The default pale-yellow correlation regions follow Wadoux, Walvoort, and Brus (2022). Points are coloured by the model-efficiency coefficient R-squared / NSE / MEC by default.

Missing pairs are removed separately per model when na.rm = TRUE. Two complete pairs with non-zero observation SD are required.

By default, point colour represents the model-efficiency coefficient (uppercase R-squared), equivalent to NSE and MEC in modelskill. This must not be confused with squared Pearson correlation, available with colour_by = "r2". The default legend title for the efficiency colour scale is simply R^2.

The function returns an ordinary ggplot2 object. Styling that belongs to the ggplot2 ecosystem can therefore be applied after the function call. Use labs() for titles, theme() for typography and legend placement, scale_colour_*() to replace the point-colour scale, and scale_fill_*() to replace the correlation-region palette.

Reference-axis arguments control the manually drawn solar axes rather than clipping limits. Observations outside those reference axes remain visible. To zoom while retaining equal x and y scaling, add a new coord_fixed() layer with the desired limits.

Value

A `ggplot2` plot object that can be extended with ordinary ggplot2 layers, scales, labels, themes, and coordinates.

Coordinates and labels

The horizontal coordinate is normalized mean error (nME; ME*) and the vertical coordinate is standardized unbiased root mean square difference (sde; SDE*). Correlation-region radii are calculated exactly from their correlation thresholds as `sqrt(1 - r^2)`; their interpretation requires the normalization assumptions documented in `diagram_stats()`.

Interpretation

The solar diagram uses the error decomposition

$$\mathrm{RMSE}^{*2} = \mathrm{ME}^{*2} + \mathrm{SDE}^{*2},$$

where ME*, SDE*, and RMSE* are respectively mean error, centred root mean square difference, and RMSE divided by the population-moment observation standard deviation. The distance from the origin is RMSE*. The origin is perfect prediction; points on the vertical axis have no mean error; negative ME* indicates overprediction under the package convention `obs - pred`; and positive ME* indicates underprediction.

Points inside the outer RMSE* = 1 circle improve on predicting the observed mean (equivalently, MEC/NSE/R2 is positive). The pale-yellow regions give lower bounds on correlation, rather than exact correlation values. For example, a point near the origin and inside the correlation-greater-than-0.9 region has low total error and strong pattern agreement; a point far left or right is systematically biased; and a point high on the vertical axis is unbiased but has substantial pattern or spread disagreement.

ggplot2 customization

Arguments that change the statistical content or core diagram construction are exposed directly by `gg_solar()`. Ordinary appearance is intentionally left to ggplot2. For example, users can add `labs()`, `theme()`, a replacement `scale_colour_*()` or `scale_fill_*()`, or a replacement `coord_fixed()`.

References

Wadoux, A. M. J.-C., Walvoort, D. J. J., and Brus, D. J. (2022). An integrated approach for the evaluation of quantitative soil maps through Taylor and solar diagrams. *Geoderma*, 405, 115332. doi:10.1016/j.geoderma.2021.115332

Jolliff, J. K., Kindle, J. C., Shulman, I., Penta, B., Friedrichs, M. A. M., Helber, R., and Arnone, R. A. (2009). Summary diagrams for coupled hydrodynamic-ecosystem model skill assessment. *Journal of Marine Systems*, 76, 64-82. doi:10.1016/j.jmarsys.2008.05.014

See Also

`diagram_stats()`, `model_metrics()`, `gg_taylor()`, `gg_target()`

Examples

```
obs <- c(1, 2, 3, 4, 5)
mods <- list(perfect = obs, biased = obs + 1)

# Default: point colour represents R-squared / NSE / MEC.
gg_solar(mods, obs)

# Give every model a categorical colour and a model-name legend.
p_models <- gg_solar(mods, obs, colour_by = "model")

# Write model names directly beside points.
p_labels <- gg_solar(mods, obs, label = TRUE)

# Standard ggplot2 customization.
p_custom <- gg_solar(mods, obs) +
  ggplot2::labs(title = "Model performance") +
  ggplot2::theme(legend.position = "bottom")
# Print p_models, p_labels or p_custom to display a variant.
```

---

gg_target *Create a target diagram*

Description

Creates a target diagram comparing quantitative prediction models with an observation vector. The horizontal coordinate is the signed standardized unbiased root mean square difference and the vertical coordinate is normalized mean error. The dashed inner circles delimit regions implying minimum Pearson correlation levels, whereas the solid unit circle provides an RMSE* reference.

Usage

```
gg_target(
  mods,
  obs,
  colorval = NULL,
  colorval.name = NULL,
  colour_by = c("efficiency", "model", "correlation", "r2"),
  axis_begin = -1.5,
  axis_end = 1.5,
  by = 0.1,
  label = FALSE,
  point_size = 7,
  label_size = 4,
  na.rm = TRUE,
  legend = TRUE,
  reference = TRUE
)
```

Arguments

mods	Numeric vector, list of numeric vectors, or numeric matrix/data frame with one model per column. Rows match obs in order. Supplied model names must be unique; missing names are generated.
obs	A numeric observation vector.
colorval	Optional finite numeric vector with one value per model, in model order. When supplied, it overrides the continuous metric selected by colour_by.
colorval.name	Optional point-colour legend title. The title can also be replaced afterwards with labs(fill = ...).
colour_by	Character string defining point colouring. "efficiency" uses uppercase R-squared / NSE / MEC and is the default; "model" gives every model a categorical colour and model-name legend; "correlation" uses Pearson correlation; and "r2" uses squared Pearson correlation.
axis_begin	Lower endpoint of both manually drawn reference axes. Defaults to -1.5.
axis_end	Upper endpoint of both manually drawn reference axes. Defaults to 1.5.
by	Spacing between manually drawn reference-axis ticks.
label	Logical; draw model names beside points using ggrepel?
point_size	Numeric point size.
label_size	Numeric size of model labels.
na.rm	Logical; remove incomplete observation-prediction pairs separately for each model? If FALSE, missing pairs cause an error because diagram coordinates cannot be calculated. The default is TRUE.
legend	Logical; show the point-colour legend?
reference	Logical; draw the target-diagram reference circles and their labels? The dashed inner circles indicate regions implying minimum Pearson correlation levels, whereas the solid unit circle represents the RMSE* = 1 reference.

Details

The default geometry follows Wadoux, Walvoort, and Brus (2022). By default, points are coloured by the model-efficiency coefficient R-squared, equivalent to NSE and MEC in modelskill.

Missing pairs are removed separately per model when na.rm = TRUE. Two complete pairs with non-zero observation standard deviation are required.

The default point-colour variable is the uppercase modelskill R2(), which is equivalent to NSE and MEC. This should not be confused with lowercase r2(), which is squared Pearson correlation.

The function returns an ordinary ggplot2 object. Styling that belongs to the ggplot2 ecosystem can therefore be applied after the function call. Use labs() for titles, theme() for typography and legend placement, and scale_fill_*() to replace the point-colour scale.

The axis arguments control the manually drawn target-diagram reference axes rather than clipping limits. Models outside those reference axes remain visible. Equal coordinate scaling is retained so that the reference circles remain circular.

Value

A ggplot2 object that can be extended with ordinary ggplot2 layers, scales, labels, themes, and coordinates.

Coordinates and labels

The horizontal coordinate is signed_sde and the vertical coordinate is nME. To retain the presentation of the original implementation, the displayed horizontal title is ME* and the displayed vertical title is SDE* multiplied by the sign of the standard-deviation difference.

Interpretation

The target diagram combines the solar error decomposition

$$\mathrm{RMSE}^{*2} = \mathrm{ME}^{*2} + \mathrm{SDE}^{*2}$$

with the sign of the prediction-versus-observation standard-deviation difference. The signed SDE coordinate distinguishes predictions with less variation than observations from predictions with greater variation; the mean-error coordinate distinguishes overprediction (negative ME*) from underprediction (positive ME*) under the obs - pred convention. The distance to the origin is RMSE normalized by the population-moment observation standard deviation, so points near the origin are preferred.

The reference circles have two distinct interpretations. The dashed inner circles correspond to correlation lower-bound regions, with radius $\sqrt{1 - r^2}$ for the displayed Pearson correlation threshold. The solid unit circle instead represents the RMSE* = 1 reference. A point near the origin is both close in mean and in spread/pattern; a point displaced along the mean-error direction is chiefly biased; and a point displaced along the signed-SDE direction chiefly differs in variability or pattern. The displayed titles intentionally retain the original implementation's visual orientation; use the coordinate definitions above when interpreting position.

ggplot2 customization

Arguments that change the statistical content or core target-diagram construction are exposed directly by gg_target(). Ordinary appearance is intentionally left to ggplot2. For example, users can add labs(), theme(), a replacement scale_fill_*(), or a replacement coord_fixed().

References

Wadoux, A. M. J.-C., Walvoort, D. J. J., and Brus, D. J. (2022). An integrated approach for the evaluation of quantitative soil maps through Taylor and solar diagrams. *Geoderma*, 405, 115332. doi:10.1016/j.geoderma.2021.115332

Jolliff, J. K., Kindle, J. C., Shulman, I., Penta, B., Friedrichs, M. A. M., Helber, R., and Arnone, R. A. (2009). Summary diagrams for coupled hydrodynamic-ecosystem model skill assessment. *Journal of Marine Systems*, 76, 64-82. doi:10.1016/j.jmarsys.2008.05.014

See Also

diagram_stats(), model_metrics(), gg_taylor(), gg_solar()

Examples

```
obs <- c(1, 2, 3, 4, 5)
mods <- list(perfect = obs, biased = obs + 1)

# Default: point colour represents R-squared / NSE / MEC.
gg_target(mods, obs)

# Give every model a categorical colour and a model-name legend.
p_models <- gg_target(mods, obs, colour_by = "model")

# Write model names directly beside points.
p_labels <- gg_target(mods, obs, label = TRUE)

# Standard ggplot2 customization.
p_custom <- gg_target(mods, obs) +
  ggplot2::labs(title = "Model performance") +
  ggplot2::theme(legend.position = "bottom")
# Print p_models, p_labels or p_custom to display a variant.
```

---

gg_taylor *Create a Taylor diagram*

Description

Creates a Taylor diagram comparing one or more quantitative prediction models with an observation vector. The radial coordinate is the model standard deviation divided by the observation standard deviation and the polar angle represents the Pearson correlation. Optional centred root-mean-square distance (RMSD) contours are drawn around the observation reference point.

Usage

```
gg_taylor(
  mods,
  obs,
  label = FALSE,
  legend = FALSE,
  point_size = 6,
  label_size = 4,
  na.rm = TRUE,
  half = FALSE,
  rmsd = TRUE,
  rmsd_colour = "red3",
  rmsd_breaks = NULL
)
```

Arguments

mods	Numeric vector, list of numeric vectors, or numeric matrix/data frame with one model per column. Rows match obs in order. Supplied model names must be unique; missing names are generated.
obs	A numeric observation vector.
label	Logical; draw model names directly on the diagram using ggrepel?
legend	Logical; colour model points by model and display a legend? The standard ggplot2 discrete colour palette is used by default.
point_size	Numeric size of model points.
label_size	Numeric text size for model labels.
na.rm	Logical; remove incomplete observation-prediction pairs separately for each model? If FALSE, missing pairs cause an error because diagram coordinates cannot be calculated. The default is TRUE.
half	Logical; if TRUE, draw only the positive-correlation portion of the Taylor diagram (r = 0 to r = 1). If FALSE, draw the full Taylor diagram (r = -1 to r = 1).
rmsd	Logical; show centred RMSD contours and their labels?
rmsd_colour	Character string giving the colour of the RMSD contours and labels.
rmsd_breaks	Optional numeric vector giving the RMSD contour values. If NULL, contours are drawn every 0.5 units.

Details

The geometry and default styling follow Wadoux, Walvoort, and Brus (2022) and the original implementation in the accompanying repository.

Missing pairs are removed separately per model when na.rm = TRUE. Two complete pairs with non-zero observation SD are required. All plotted statistics can be retrieved with diagram_stats().

Model names can be displayed either directly on the diagram with label = TRUE or through a colour legend with legend = TRUE.

With legend = TRUE, model points use the standard ggplot2 discrete colour palette. Because the returned object is a regular ggplot2 object, the model colour scale, legend, theme, titles, fonts, and other graphical elements can subsequently be customised with ordinary ggplot2 layers.

RMSD contours can be removed with rmsd = FALSE, recoloured with rmsd_colour, or placed at user-defined values with rmsd_breaks.

Value

A ggplot2 plot object. It can be extended with ordinary ggplot2 layers, scales, labels, and themes.

Interpretation

Let sigma-star denote the prediction standard deviation divided by the observation standard deviation, and let r be Pearson correlation. The Taylor geometry follows

$$\mathrm{SDE}^* = \sqrt{1 + \sigma_{\mathrm{star}}^2 - 2\sigma_{\mathrm{star}} r},$$

where SDE* is the centred (unbiased) root mean square difference divided by the observation standard deviation. Radial distance gives `sigma_star`; the polar angle is `acos(r)`; and the reference point has a standard-deviation ratio of one and correlation of one. Points nearer the reference point have smaller unbiased error.

A point inside the unit-radius arc has less variation than the observations (a smoother prediction), while a point outside it has greater variation. Points nearer the horizontal positive-correlation axis have stronger pattern agreement. The diagram does not show mean error: a model can be close to the reference point but systematically biased. Use `gg_solar()` or `gg_target()` together with `bias()` when mean error is important.

References

Wadoux, A. M. J.-C., Walvoort, D. J. J., and Brus, D. J. (2022). An integrated approach for the evaluation of quantitative soil maps through Taylor and solar diagrams. *Geoderma*, 405, 115332. doi:10.1016/j.geoderma.2021.115332

Taylor, K. E. (2001). Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research*, 106, 7183-7192. doi:10.1029/2000JD900719

See Also

`diagram_stats()`, `model_metrics()`, `gg_solar()`, `gg_target()`

Examples

```
obs <- c(1, 2, 3, 4, 5)

mods <- list(
  perfect = obs,
  biased = obs + 1,
  smooth = c(2, 2, 3, 4, 4)
)

# Positive-correlation Taylor diagram with model names
gg_taylor(mods, obs, label = TRUE, half = TRUE)
```

---

interval_score *Central prediction-interval score*

Description

The interval score is `upper - lower + 2 / alpha * (lower - obs)` below the interval and `upper - lower + 2 / alpha * (obs - upper)` above it, where `alpha = 1 - level`; there is no penalty inside the interval. Lower scores indicate sharper, well-calibrated intervals.

Usage

```
interval_score(
  obs,
  lower = NULL,
  upper = NULL,
  level = 0.95,
  na.rm = TRUE,
  pred = NULL,
  predictive_sd = NULL,
  distribution = NULL
)
```

Arguments

`obs`	Numeric observation vector.
`lower`, `upper`	Numeric bounds of the central prediction interval. For the usual proper-score interpretation, these must be the equal-tailed predictive quantiles corresponding to `level`.
`level`	Nominal central interval coverage, strictly between zero and one; determines the required predictive quantiles for `lower` and `upper`.
`na.rm`	Logical; remove incomplete triplets?
`pred`, `predictive_sd`	Optional numeric vectors of predictive means and predictive standard deviations, supplied together and of the same length as `obs`. Assumes normal predictive distributions. Non-missing predictive standard deviations must be finite and strictly positive.
`distribution`	Optional numeric matrix or data frame of equally weighted predictive samples: one row per observation and one column per predictive draw. Supply this instead of bounds or predictive means and standard deviations. At least one draw is required; infinite values are not allowed.

Details

For the usual proper-scoring interpretation, `lower` and `upper` must be the central (equal-tailed) predictive quantiles corresponding to `level`: the predictive quantiles at probabilities `(1 - level) / 2` and `(1 + level) / 2`, respectively. Supplying arbitrary interval bounds together with a nominal `level` still evaluates the formula below, but does not in general give the same proper interval-score interpretation.

$$\mathrm{IS}_\tau = \frac{1}{n}\sum_{i=1}^{n}[upper_i - lower_i + \frac{2}{1-\tau}\max(lower_i - obs_i, 0) + \frac{2}{1-\tau}\max(obs_i - upper_i, 0)]$$

The score has response units and lower values are better. It rewards narrow intervals but penalizes observations outside them by their distance from the nearest bound.

Value

One numeric value.

Input representations

Supply exactly one of explicit `lower` and `upper` bounds, predictive mean and standard deviation (`pred` and `predictive_sd`), or predictive samples (`distribution`). Normal inputs generate central intervals using normal quantiles. Samples generate equal-tailed intervals using `stats::quantile()` with `type = 7`, at probabilities `(1 - level) / 2` and `(1 + level) / 2`. With `na.rm = TRUE`, a case is removed if its observation or any supplied predictive value is missing; individual missing draws are not discarded within a case. With `na.rm = FALSE`, incomplete inputs give `NA`. No complete cases also gives `NA`. Standalone `interval_width()` ignores missing `obs`.

References

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102, 359-378.

Examples

```
interval_score(1:3, c(0, 1, 2), c(2, 3, 4), level = .8)
```

---

interval_width *Average prediction-interval width*

Description

Arithmetic mean of `upper - lower`, i.e. PIW(tau) = sum(upper - lower) / n for a tau-level prediction interval. Smaller widths are sharper, but should always be interpreted jointly with empirical coverage. PIW is independent of observed values: `obs` is retained only to check input length compatibility. Central intervals have lower and upper predictive quantiles at `(1 - tau) / 2` and `(1 + tau) / 2`, respectively.

Usage

```
interval_width(
  obs,
  lower = NULL,
  upper = NULL,
  na.rm = TRUE,
  level = 0.95,
  pred = NULL,
  predictive_sd = NULL,
  distribution = NULL
)
```

Arguments

`obs`	Numeric observation vector.
`lower`, `upper`	Optional numeric lower and upper prediction-interval bounds. Supply both, without another input representation.
`na.rm`	Logical; remove incomplete cases? See Input representations.
`level`	Nominal central interval coverage, strictly between zero and one. Used to generate bounds from predictive means and standard deviations or predictive samples; it does not alter explicit bounds.
`pred`, `predictive_sd`	Optional numeric vectors of predictive means and predictive standard deviations, supplied together and of the same length as `obs`. Assumes normal predictive distributions. Non-missing predictive standard deviations must be finite and strictly positive.
`distribution`	Optional numeric matrix or data frame of equally weighted predictive samples: one row per observation and one column per predictive draw. Supply this instead of bounds or predictive means and standard deviations. At least one draw is required; infinite values are not allowed.

Details

$$\mathrm{PIW}(\tau) = \frac{1}{n}\sum_{i=1}^{n}(upper_i - lower_i)$$

PIW has response units. Smaller values indicate sharper predictions, but are desirable only when calibration is adequate; assess it alongside `picp()`.

Value

One numeric value.

Input representations

Supply exactly one of explicit `lower` and `upper` bounds, predictive mean and standard deviation (`pred` and `predictive_sd`), or predictive samples (`distribution`). Normal inputs generate central intervals using normal quantiles. Samples generate equal-tailed intervals using `stats::quantile()` with `type = 7`, at probabilities `(1 - level) / 2` and `(1 + level) / 2`. With `na.rm = TRUE`, a case is removed if its observation or any supplied predictive value is missing; individual missing draws are not discarded within a case. With `na.rm = FALSE`, incomplete inputs give `NA`. No complete cases also gives `NA`. Standalone `interval_width()` ignores missing `obs`.

References

Schmidinger, J. and Heuvelink, G. B. M. (2023). Validation of uncertainty predictions in digital soil mapping. *Geoderma*, 437, 116585. doi:10.1016/j.geoderma.2023.116585

Examples

```
interval_width(1:3, c(0, 1, 2), c(2, 3, 4))
```

---

kge *Kling-Gupta efficiency: KGE (2009)*

Description

KGE (2009) is the original Kling-Gupta efficiency formulation of Gupta et al. (2009). It combines correlation, variability ratio, and mean ratio. The component definitions below use the same paired observations and predictions as the main score. Later KGE variants use different component definitions and are not implemented by this function.

Usage

```
kge(obs, pred, na.rm = TRUE)
```

Arguments

obs	Numeric observation vector.
pred	Numeric prediction vector paired with obs.
na.rm	Logical; remove incomplete pairs?

Details

$$r = \frac{\sum_{i=1}^{n}(obs_i - \bar{obs})(pred_i - \bar{pred})}{\sqrt{\sum_{i=1}^{n}(obs_i - \bar{obs})^2 \sum_{i=1}^{n}(pred_i - \bar{pred})^2}}, \quad \alpha = \sqrt{\frac{\sum_{i=1}^{n}(pred_i - \bar{pred})^2}{\sum_{i=1}^{n}(obs_i - \bar{obs})^2}}, \quad \beta = \frac{\bar{pred}}{\bar{obs}}.$$

$$\mathrm{KGE}_{2009} = 1 - \sqrt{(r-1)^2 + (\alpha - 1)^2 + (\beta - 1)^2}.$$

One is ideal. Values closer to one indicate agreement in linear association, spread, and mean. The range is unbounded below and at most one. Zero is not the observed-mean benchmark used for NSE. KGE (2009) is undefined when the observed mean or either vector's standard deviation is zero, or fewer than two pairs remain; it returns NA with a warning in those cases. As with NSE, avoid treating KGE (2009) as the only measure of model quality; inspect its components and complementary error metrics.

Value

One numeric KGE (2009) value; one is ideal.

References

Gupta, H. V., Kling, H., Yilmaz, K. K., and Martinez, G. F. (2009). Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *Journal of Hydrology*, 377, 80-91. doi:10.1016/j.jhydrol.2009.08.003

`log_score` *Logarithmic (ignorance) score*

Description

Returns the mean negative log predictive density at the observations. Lower values are better. This score requires positive predictive densities and is particularly sensitive to observations assigned very low density.

Usage

```
log_score(obs, density_at_obs, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector, retained for length checking.
`density_at_obs`	Numeric vector of strictly positive predictive-density values evaluated at each corresponding observation.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\text{Log score} = -\frac{1}{n}\sum_{i=1}^{n}\log f_i(obs_i)$$

Lower values are better. The score strongly penalizes assigning near-zero density to observations, so it is useful for comparing full predictive distributions but can be dominated by tail failures.

Value

One numeric score.

References

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *JASA*, 102, 359-378. doi:10.1198/016214506000001437

Examples

```
log_score(0, stats::dnorm(0))
```

`mae` *Mean absolute error*

Description

Mean absolute error (MAE) is the average absolute difference between observations and predictions.

Usage

```
mae(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|obs_i - pred_i|$$

MAE is non-negative and has the same units as the response variable. Zero indicates perfect predictions; smaller values indicate smaller typical prediction errors. Unlike mean error (ME), positive and negative errors cannot cancel. MAE gives each error equal weight and is therefore less sensitive to unusually large errors than `rmse()`. Interpret MAE alongside `bias()` to assess both typical error magnitude and systematic over- or underprediction. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Willmott, C. J. and Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30, 79-82. doi:10.3354/cr030079

Hodson, T. O. (2022). Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development*, 15, 5481-5487. doi:10.5194/gmd-15-5481-2022

Examples

```
mae(1:3, c(1, 3, 2))
```

mape *Mean absolute percentage error*

Description

Mean absolute percentage error (MAPE) averages absolute error relative to each observation.

Usage

```
mape(obs, pred, na.rm = TRUE)
```

Arguments

obs	Numeric observation vector.
pred	Numeric prediction vector paired with obs.
na.rm	Logical; remove incomplete pairs?

Details

$$\mathrm{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{obs_i - pred_i}{obs_i} \right|.$$

MAPE is a non-negative percentage; zero is ideal. It returns NA with a warning when any observation is zero and can disproportionately weight errors near zero. Missing-value handling follows bias().

Value

One numeric value.

References

Hyndman and Koehler (2006). See mdae().

See Also

Other prediction metrics: mdae(), mpe(), msle(), rae(), rer(), rmsle(), rpd(), rpiq(), rrmse(), sep(), smape(), willmott_d()

mdae *Median absolute error*

Description

Median absolute error (MdAE) is the median absolute prediction error.

Usage

```
mdae(obs, pred, na.rm = TRUE)
```

Arguments

obs	Numeric observation vector.
pred	Numeric prediction vector paired with obs.
na.rm	Logical; remove incomplete pairs?

Details

$$\mathrm{MdAE} = \mathrm{median}_{i=1,\ldots,n}(|obs_i - pred_i|).$$

MdAE has response units, is non-negative, and zero is ideal. It describes a typical error while being less sensitive to extreme errors than `mae()` or `rmse()`. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22, 679-688. doi:10.1016/j.ijforecast.2006.03.001

See Also

Other prediction metrics: `mape()`, `mpe()`, `msle()`, `rae()`, `rer()`, `rmsle()`, `rpd()`, `rpiq()`, `rrmse()`, `sep()`, `smape()`, `willmott_d()`

mec *Model efficiency coefficient*

Description

Alias for `nse()`. MEC, NSE, and the uppercase R-squared efficiency `R2()` are the same statistic. They must not be confused with lowercase `r2()`, the squared Pearson correlation.

Usage

```
mec(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\mathrm{MEC} = 1 - \frac{\sum_{i=1}^{n}(obs_i - pred_i)^2}{\sum_{i=1}^{n}(obs_i - \bar{obs})^2}.$$

Its interpretation and references are given in `nse()`. It returns `NA` with a warning under the same undefined conditions as `nse()`.

Value

One numeric value.

Examples

```
mec(1:3, c(1, 3, 2))
```

median_crps *Median continuous ranked probability score*

Description

Median of case-wise CRPS values. Lower values are better. It is a robust descriptive summary when a few large errors dominate mean CRPS.

Usage

```
median_crps(
  obs,
  distribution = NULL,
  pred = NULL,
  predictive_sd = NULL,
  na.rm = TRUE
)
```

Arguments

`obs`	Numeric observation vector.
`distribution`	Numeric matrix/data frame of equally weighted predictive samples, one row per observation.
`pred`, `predictive_sd`	Mean and strictly positive predictive SD for normal predictive distributions.
`na.rm`	Logical; remove incomplete observation/distribution rows?

Details

$$\text{median CRPS} = \text{median}(\text{CRPS}_i)$$

It has response units; lower values are better. Unlike mean CRPS, it describes a typical case and is less sensitive to a small number of very poor predictive distributions. Median aggregation is a robust descriptive summary but should not replace mean CRPS for formal comparisons based on proper scoring rules.

Value

One numeric median CRPS value.

References

Hersbach, H. (2000). Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, 15, 559-570. doi:10.1175/1520-0434(2000)015%3C0559:DOTCRP%3E2.0.CO;2

Examples

```
median_crps(0, distribution = matrix(c(-1, 1), nrow = 1))
```

`model_metrics` *Calculate prediction-validation metrics for one or more models*

Description

Applies the individual prediction metrics to one or more prediction vectors. Each statistic is returned once. Efficiency is named `R2`; standalone `nse()` and `mec()` remain aliases. See the linked metric help pages for equations, references, and interpretation.

Usage

```
model_metrics(mods, obs, na.rm = TRUE, extended = FALSE, digits = NULL)
```

Arguments

`mods`	Numeric vector, list of numeric vectors, or numeric matrix/data frame with one model per column. Rows match `obs` in order. Supplied model names must be unique; missing names are generated.
`obs`	A numeric observation vector.
`na.rm`	Logical; whether incomplete observation-prediction pairs should be removed. The default is `TRUE`. If `FALSE`, incomplete pairs result in missing statistics rather than being silently removed.
`extended`	Logical; include additional robust, scale-normalised, percentage, agreement, and KGE (2009) metrics?
`digits`	Integer or `NULL`. If supplied, round numeric results to this many digits (0 to 22). `NULL` retains full numerical precision.

Details

Errors are observation minus prediction. See `bias()`, `crmse()`, `nse()`, and `ccc()` for definitions and interpretation.

Errors are observation minus prediction: negative ME indicates overprediction. ME, MAE and RMSE have the input units. NSE is one minus the ratio of squared error to the observation sum of squares about its mean. Zero is the observation-mean benchmark and negative values are worse. Concordance uses population variances (divisor n); rhoC = r * Cb. Missing pairs are removed separately for each model, so comparisons may use different subsets. NA and NaN are missing; infinite values are rejected. Repeated observations are retained with equal weight. With fewer than two pairs, correlation and efficiency are NA. Constant inputs return NA for r and r2 because Pearson correlation is undefined. Constant observations give NA efficiency. If either vector is constant, Cb and rhoC are zero when the concordance denominator is positive; both are NA for identical constant vectors (zero denominator). All-missing models return NA.

Value

A base data frame with one row per model and canonical columns `model`, `bias`, `mae`, `mse`, `rmse`, `nrmse`, `crmse`, `correlation`, `r2`, `R2`, `sd_ratio`, `ccc`, and `Cb`. With `extended = TRUE`, adds `mdae`, `rpd`, `rpiq`, `sep`, `rer`, `mape`, `mpe`, `smape`, `msle`, `rmsle`, `rae`, `rrmse`, `willmott_d`, and `kge` (KGE (2009)). Duplicate columns ME, MAE, RMSE, r, nse, NSE, MEC, and rhoC have been removed; use bias, mae, rmse, correlation, R2, and ccc instead.

Bias correction factor

`Cb` is Lin's bias correction factor, using population SDs and means:

$$C_b = \frac{2\sigma_o\sigma_p}{\sigma_o^2 + \sigma_p^2 + (\mu_o - \mu_p)^2}.$$

It ranges from zero to one; one indicates equal means and SDs. It does not measure correlation. For defined correlation, CCC equals r times Cb. See `ccc()` and Lin (1989), doi:10.2307/2532051.

References

Wadoux, A. M. J.-C., Walvoort, D. J. J., and Brus, D. J. (2022). An integrated approach for the evaluation of quantitative soil maps through Taylor and solar diagrams. *Geoderma*, 405, 115332. doi:10.1016/j.geoderma.2021.115332

See Also

`diagram_stats()`, `gg_taylor()`, `gg_solar()`, `gg_target()`

Examples

```
obs <- c(1, 2, 3, 4, 5)
preds <- list(model_a = c(1, 2, 3, 4, 5),
              model_b = c(2, 2, 3, 4, 4))
model_metrics(preds, obs)
```

---

mpe *Mean percentage error*

Description

Mean percentage error (MPE) is signed mean error relative to observations, reported in percent.

Usage

```
mpe(obs, pred, na.rm = TRUE)
```

Arguments

obs	Numeric observation vector.
pred	Numeric prediction vector paired with obs.
na.rm	Logical; remove incomplete pairs?

Details

$$\mathrm{MPE} = \frac{100}{n}\sum_{i=1}^{n}\frac{obs_i - pred_i}{obs_i}.$$

MPE is unbounded and zero is ideal. For strictly positive observations, positive values indicate underprediction. Relative errors can cancel; MPE is undefined for zero observations, returning NA with a warning, and unstable near zero. Missing-value handling follows bias().

Value

One numeric value.

References

Hyndman and Koehler (2006). See mdae().

See Also

Other prediction metrics: mape(), mdae(), msle(), rae(), rer(), rmsle(), rpd(), rpiq(), rrmse(), sep(), smape(), willmott_d()

---

mse *Mean squared error*

Description

Mean squared error (MSE) is the mean squared difference between observations and predictions.

Usage

```
mse(obs, pred, na.rm = TRUE)
```

Arguments

obs	Numeric observation vector.
pred	Numeric prediction vector paired with obs.
na.rm	Logical; remove incomplete pairs?

Details

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(obs_i - pred_i)^2$$

MSE is non-negative and zero indicates perfect predictions. Smaller values indicate better agreement. Squaring gives larger errors disproportionately more influence, making MSE sensitive to large deviations. Its units are the squared units of the response variable, so `rmse()` is usually easier to interpret on the original response scale. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Hodson, T. O. (2022). Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development*, 15, 5481-5487. doi:10.5194/gmd-15-5481-2022

Examples

```
mse(1:3, c(1, 3, 2))
```

---

msle *Mean squared logarithmic error*

Description

Mean squared logarithmic error (MSLE) averages squared differences on the log1p scale.

Usage

```
msle(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\mathrm{MSLE} = \frac{1}{n}\sum_{i=1}^{n}[\log(1 + obs_i) - \log(1 + pred_i)]^2.$$

MSLE is non-negative and zero is ideal. It emphasizes relative differences and requires non-negative observations and predictions; otherwise it returns `NA` with a warning. The log1p convention is a package choice. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Hodson, T. O. (2022). Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development*, 15, 5481-5487. doi:10.5194/gmd-15-5481-2022

See Also

Other prediction metrics: `mape()`, `mdae()`, `mpe()`, `rae()`, `rer()`, `rmsle()`, `rpd()`, `rpiq()`, `rrmse()`, `sep()`, `smape()`, `willmott_d()`

`nrmse` *Normalized root mean squared error*

Description

Normalized RMSE (NRMSE) divides `rmse()` by the sample standard deviation of observations.

Usage

```
nrmse(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\mathrm{NRMSE} = \frac{\sqrt{n^{-1}\sum_{i=1}^{n}(obs_i - pred_i)^2}}{\sqrt{(n-1)^{-1}\sum_{i=1}^{n}(obs_i - \bar{obs})^2}}.$$

NRMSE is unitless and zero is ideal. Smaller values indicate less error relative to the observed variation. A value of one means RMSE equals one observed sample standard deviation. This package uses this normalization to remain consistent with its diagram statistics. It returns `NA` with a warning when fewer than two valid pairs remain or the observations have zero variance. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Taylor, K. E. (2001). Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research*, 106, 7183-7192. doi:10.1029/2000JD900719

Examples

```
nrmse(1:3, c(1, 3, 2))
```

`nse` *Nash-Sutcliffe efficiency*

Description

Nash-Sutcliffe efficiency (NSE; also called the model efficiency coefficient, MEC) compares the prediction squared error with the squared error from using the observed mean as a constant prediction.

Usage

```
nse(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{n}(obs_i - pred_i)^2}{\sum_{i=1}^{n}(obs_i - \bar{obs})^2}$$

One is ideal; zero means the predictions are no better than predicting the observed mean; negative values indicate worse performance than that benchmark. NSE returns `NA` with a warning when fewer than two valid pairs remain or the observations have zero variance. It is sensitive to large errors because it uses squared differences. NSE, `mec()`, and uppercase `R2()` are identical in this package; they are not lowercase `r2()`.

Value

One numeric value.

References

Nash, J. E. and Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I: A discussion of principles. *Journal of Hydrology*, 10, 282-290. doi:10.1016/0022-1694(70)90255-6

Janssen, P. H. M. and Heuberger, P. S. C. (1995). Calibration of process-oriented models. *Ecological Modelling*, 83, 55-66. doi:10.1016/0304-3800(95)00084-9

Legates, D. R. and McCabe, G. J. (1999). Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation. *Water Resources Research*, 35(1), 233-241. doi:10.1029/1998WR900018

Examples

```
nse(1:3, c(1, 3, 2))
```

---

picp *Prediction interval coverage probability (PICP)*

Description

PICP is the empirical proportion of observations satisfying `lower <= obs <= upper`; endpoints are included. It is returned on the probability scale from zero to one, so multiply by 100 to report a percent.

Usage

```
picp(
  obs,
  lower = NULL,
  upper = NULL,
  na.rm = TRUE,
  level = 0.95,
  pred = NULL,
  predictive_sd = NULL,
  distribution = NULL
)
```

Arguments

`obs`	Numeric observation vector.
`lower, upper`	Optional numeric lower and upper prediction-interval bounds. Supply both, without another input representation.
`na.rm`	Logical; remove incomplete cases? See Input representations.
`level`	Nominal central interval coverage, strictly between zero and one. Used to generate bounds from predictive means and standard deviations or predictive samples; it does not alter explicit bounds.

pred, predictive_sd
: Optional numeric vectors of predictive means and predictive standard deviations, supplied together and of the same length as obs. Assumes normal predictive distributions. Non-missing predictive standard deviations must be finite and strictly positive.

distribution
: Optional numeric matrix or data frame of equally weighted predictive samples: one row per observation and one column per predictive draw. Supply this instead of bounds or predictive means and standard deviations. At least one draw is required; infinite values are not allowed.

Details

$$\mathrm{PICP}(\tau) = \frac{1}{n}\sum_{i=1}^{n} I(lower_i \le obs_i \le upper_i)$$

For a well-calibrated central interval, picp() should be close to its nominal level. Missing triplets are removed when na.rm = TRUE.

A single PICP does not reveal whether non-coverage is balanced between the lower and upper tails. Use gg_coverage() to inspect PICP across interval levels; assess tail-specific calibration separately when directional bias is scientifically important.

Value

One numeric value.

Input representations

Supply exactly one of explicit lower and upper bounds, predictive mean and standard deviation (pred and predictive_sd), or predictive samples (distribution). Normal inputs generate central intervals using normal quantiles. Samples generate equal-tailed intervals using stats::quantile() with type = 7, at probabilities (1 - level) / 2 and (1 + level) / 2. With na.rm = TRUE, a case is removed if its observation or any supplied predictive value is missing; individual missing draws are not discarded within a case. With na.rm = FALSE, incomplete inputs give NA. No complete cases also gives NA. Standalone interval_width() ignores missing obs.

References

Goovaerts, P. (2001). Geostatistical modelling of uncertainty in soil science. *Geoderma*, 103, 3-26. doi:10.1016/S0016-7061(01)00067-2

Shrestha, D. L. and Solomatine, D. P. (2008). Data-driven approaches for estimating uncertainty in rainfall-runoff modelling. *International Journal of River Basin Management*, 6, 109-122. doi:10.1080/15715124.2008.9635341

Examples

```
picp(1:3, c(0, 1, 2), c(2, 3, 4))
picp(1:3, pred = 1:3, predictive_sd = rep(1, 3), level = 0.8)
picp(1:3, distribution = cbind(0:2, 1:3, 2:4), level = 0.8)
```

`pinball_loss` *Quantile (pinball) loss*

Description

Quantile loss evaluates a prediction for a specified conditional quantile. With quantile level `tau`, it is

Usage

```
pinball_loss(obs, pred, level = 0.5, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`level`	Quantile level strictly between zero and one.
`na.rm`	Logical; remove incomplete pairs?

Details

$$L_\tau = \frac{1}{n}\sum_{i=1}^{n}\begin{cases}\tau(obs_i - pred_i), & obs_i - pred_i \geq 0\\ (\tau-1)(obs_i - pred_i), & obs_i - pred_i < 0.\end{cases}$$

It is non-negative and zero is ideal. Underprediction is penalized more when `level` is high; overprediction is penalized more when `level` is low. At `level = 0.5`, it equals one-half of `mae()`. Missing-value handling follows `bias()`. It returns `NA` with a warning when no valid pairs remain.

Value

One numeric loss; lower is better.

References

Koenker, R. and Bassett, G. (1978). Regression quantiles. *Econometrica*, 46, 33-50. doi:10.2307/1913643

`pit` *Probability integral transform*

Description

Calculates probability integral transform (PIT) values for continuous predictive distributions. PIT values can either be supplied directly as predictive cumulative distribution function (CDF) values evaluated at the observations, or calculated from predictive means and standard deviations under a normal predictive-distribution assumption. Predictive samples may instead be supplied through `distribution`.

Usage

```
pit(
  cdf_at_obs = NULL,
  obs = NULL,
  pred = NULL,
  predictive_sd = NULL,
  na.rm = TRUE,
  distribution = NULL
)
```

Arguments

`cdf_at_obs`	Optional numeric vector containing predictive CDF values evaluated at the corresponding observations. Values must lie between zero and one. Do not supply this together with `obs`, `pred`, or `predictive_sd`, or `distribution`.
`obs`	Optional numeric vector of observations. Must be supplied together with `pred` and `predictive_sd`, or with `distribution`.
`pred`	Optional numeric vector of predictive means. Used with `obs` and `predictive_sd` to calculate PIT values assuming normal predictive distributions.
`predictive_sd`	Optional numeric vector of predictive standard deviations. Values must be finite and strictly positive.
`na.rm`	Logical; remove incomplete values or observation-prediction combinations? The default is `TRUE`.
`distribution`	Optional numeric matrix or data frame of equally weighted predictive samples: one row per observation and one column per predictive draw. Supply this instead of bounds or predictive means and standard deviations. At least one draw is required; infinite values are not allowed.

Details

For observation obs_i and predictive cumulative distribution function F_i, the PIT value is

$$u_i = F_i(obs_i).$$

For calibrated continuous predictive distributions, PIT values evaluated over independent validation observations should be approximately uniformly distributed between zero and one. Individual PIT values do not have a preferred value; calibration is assessed from their distribution across observations, for example with `gg_pit()`.

When `cdf_at_obs` is supplied, the values are returned after validation. This mode can be used with any continuous predictive distribution provided its CDF has already been evaluated at each corresponding observation.

Alternatively, `obs`, `pred`, and `predictive_sd` can be supplied together. In this case, normal predictive distributions are assumed and PIT values are calculated as

$$u_i = \Phi\left(\frac{obs_i - pred_i}{\sigma_i}\right),$$

where Φ is the standard normal CDF and σ_i is the predictive standard deviation.

A uniform PIT distribution is consistent with probabilistic calibration. Systematic departures from uniformity can indicate misspecification of the predictive distributions. For example, U-shaped PIT histograms are commonly associated with underdispersed predictive distributions, hump-shaped histograms with overdispersed distributions, and asymmetric PIT histograms with systematic bias. These patterns are diagnostic rather than unique and should be interpreted together with other validation measures.

PIT uniformity assesses the predictive distributions collectively and does not by itself establish that every aspect of conditional calibration is correct. Calibration should therefore generally be evaluated using complementary diagnostics such as `gg_qcp()`, `gg_coverage()`, and proper scoring rules.

The usual uniformity interpretation applies directly to continuous predictive distributions. For discrete distributions or finite predictive ensembles, ordinary PIT values are discrete; randomized PIT or rank-based diagnostics are more appropriate.

With predictive samples, `pit()` returns the fraction of draws less than or equal to each observation (including ties). This empirical CDF requires no normal assumption. With `na.rm = TRUE`, rows missing `obs` or any draw are removed together; with `na.rm = FALSE`, an incomplete row makes all returned PIT values `NA`, retaining the original observation-vector length.

Value

Numeric vector of PIT values between zero and one.

References

Gneiting, T., Balabdaoui, F. and Raftery, A. E. (2007). Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B*, 69, 243-268. doi:10.1111/j.1467-9868.2007.00587.x

Schmidinger, J. and Heuvelink, G. B. M. (2023). Validation of uncertainty predictions in digital soil mapping. *Geoderma*, 437, 116585. doi:10.1016/j.geoderma.2023.116585

See Also

`gg_pit()`, `qcp()`, `gg_qcp()`, `picp()`, `gg_coverage()`, `crps()`

Examples

```
# PIT values from CDF values calculated elsewhere
cdf_values <- stats::pnorm(c(-1, 0, 1))
pit(cdf_values)

# Calculate PIT directly for normal predictive distributions
set.seed(123)

n <- 500
pred <- seq(0, 10, length.out = n)
predictive_sd <- rep(1, n)
obs <- stats::rnorm(n, mean = pred, sd = predictive_sd)

pit_values <- pit(
  obs = obs,
  pred = pred,
  predictive_sd = predictive_sd
)

gg_pit(pit_values)
pit(obs = 1:3, distribution = cbind(0:2, 1:3, 2:4))
```

---

qcp *Quantile coverage probability*

Description

Quantile coverage probability (QCP) evaluates the calibration of individual predictive quantiles. For a predicted quantile at nominal probability p, QCP is the empirical proportion of observations that are less than or equal to that predicted quantile.

Usage

```
qcp(
  obs,
  quantiles = NULL,
  levels = NULL,
  na.rm = TRUE,
  pred = NULL,
  predictive_sd = NULL,
  distribution = NULL
)
```

Arguments

obs	Numeric observation vector.

`quantiles` Numeric matrix or data frame with observations in rows and predicted quantiles in columns. Supply this with `levels`, without another input representation.

`levels` Strictly increasing quantile probabilities, one per column of `quantiles`. Values must lie strictly between zero and one. For generated quantiles, defaults to `seq(0.05, 0.95, by = 0.05)` and is sorted with duplicates removed.

`na.rm` Logical; remove incomplete observation/quantile rows?

`pred`, `predictive_sd` Optional numeric vectors of predictive means and predictive standard deviations, supplied together and of the same length as `obs`. Assumes normal predictive distributions. Non-missing predictive standard deviations must be finite and strictly positive.

`distribution` Optional numeric matrix or data frame of equally weighted predictive samples: one row per observation and one column per predictive draw. Supply this instead of bounds or predictive means and standard deviations. At least one draw is required; infinite values are not allowed.

Details

$$\mathrm{QCP}(p) = \frac{1}{n}\sum_{i=1}^{n} I(obs_i \le q_{i,p})$$

For a calibrated predictive distribution, QCP should be close to the nominal quantile probability `p`. For example, approximately 5% of observations should fall below predicted 0.05 quantiles, approximately 50% below predicted medians, and approximately 95% below predicted 0.95 quantiles.

Values above `p` indicate that observations fall below the predicted quantile more frequently than expected, whereas values below `p` indicate that they do so less frequently than expected.

QCP differs from prediction interval coverage probability (`picp()`). PICP evaluates the joint coverage of a lower and upper prediction-interval bound, whereas QCP evaluates each predictive quantile separately. QCP can therefore reveal asymmetric or one-sided miscalibration that may be hidden by apparently satisfactory central prediction-interval coverage.

Quantile calibration is generally most informative when evaluated across several probability levels rather than at a single quantile. A dense sequence of quantiles provides a more complete view of distributional calibration, although selected levels remain useful when only specific predictive quantiles are available.

Instead of explicit quantiles, supply predictive mean and standard deviation (`pred` and `predictive_sd`) for normal quantiles, or `distribution` for empirical quantiles (`stats::quantile()`, `type = 7`). These representations are mutually exclusive. Complete rows are selected across `obs` and all supplied predictive values, so every level uses the same cases. If `na.rm = FALSE` and any case is incomplete, or no complete cases remain, every QCP value is `NA`.

Value

Named numeric vector containing one QCP value for each supplied quantile level, on the probability scale from zero to one.

References

Schmidinger, J. and Heuvelink, G. B. M. (2023). Validation of uncertainty predictions in digital soil mapping. *Geoderma*, 437, 116585. doi:10.1016/j.geoderma.2023.116585

See Also

gg_qcp(), picp(), gg_coverage(), pit(), gg_pit()

Examples

```
set.seed(123)

n <- 500
pred <- seq(0, 10, length.out = n)
predictive_sd <- rep(1, n)
obs <- stats::rnorm(n, mean = pred, sd = predictive_sd)

levels <- seq(0.05, 0.95, by = 0.05)

quantiles <- vapply(
  levels,
  function(p) pred + stats::qnorm(p) * predictive_sd,
  numeric(n)
)

qcp(
  obs,
  quantiles = quantiles,
  levels = levels
)
qcp(obs, pred = pred, predictive_sd = predictive_sd, levels = levels)
qcp(1:3, distribution = cbind(0:2, 1:3, 2:4), levels = c(0.25, 0.75))
```

R2 *Model-efficiency R-squared*

Description

R2() is the model-efficiency coefficient and is identical to nse() and mec() in this package. It compares the squared prediction error with the squared deviation of the observations from their mean:

Usage

```
R2(obs, pred, na.rm = TRUE)
```

Arguments

obs	Numeric observation vector.
pred	Numeric prediction vector paired with obs.
na.rm	Logical; remove incomplete pairs?

Details

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(obs_i - pred_i)^2}{\sum_{i=1}^{n}(obs_i - \bar{obs})^2}.$$

The statistic has a direct benchmark interpretation. A value of 1 indicates perfect agreement between observations and predictions. A value of 0 means that predicting the observed mean for every observation performs equally well according to squared error. Negative values indicate that the observed mean provides a better prediction than the model.

Uppercase R2() must not be confused with lowercase r2(), which is the squared Pearson correlation coefficient. Squared Pearson correlation measures the strength of linear association and is insensitive to additive and proportional differences between observations and predictions. It can therefore equal one even for strongly biased predictions. In contrast, R2() is sensitive to deviations from the line of equality and therefore measures predictive performance rather than linear association alone.

Because R2() is based on squared errors, individual large prediction errors can have a disproportionate influence on its value. It should therefore generally be interpreted together with complementary measures such as bias(), mae(), rmse(), and r2() rather than as a standalone measure of predictive performance.

R2() returns NA with a warning when fewer than two valid observation- prediction pairs remain or when the observations have zero variance. Missing-value handling follows bias().

Value

One numeric value. The optimum is 1; values may be negative and are not bounded below.

References

Wadoux, A. M. J.-C., Walvoort, D. J. J. and Brus, D. J. (2022). An integrated approach for the evaluation of quantitative soil maps through Taylor and solar diagrams. *Geoderma*, 405, 115332. doi:10.1016/j.geoderma.2021.115332

Janssen, P. H. M. and Heuberger, P. S. C. (1995). Calibration of process-oriented models. *Ecological Modelling*, 83, 55-66. doi:10.1016/0304-3800(95)00084-9

Nash, J. E. and Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I: A discussion of principles. *Journal of Hydrology*, 10, 282-290. doi:10.1016/0022-1694(70)90255-6

Legates, D. R. and McCabe, G. J. (1999). Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation. *Water Resources Research*, 35(1), 233-241. doi:10.1029/1998WR900018

See Also

r2(), nse(), mec(), bias(), mae(), rmse()

Examples

```
obs <- c(1, 2, 3, 4, 5)

# Perfect predictions
R2(obs, obs)

# Additive bias: r2 remains 1, whereas R2 decreases
pred <- obs + 1
r2(obs, pred)
R2(obs, pred)

# Predictions can perform worse than using the observed mean
R2(obs, rev(obs))
```

---

r2 *Squared Pearson correlation*

Description

Squared Pearson correlation is the square of correlation().

Usage

```
r2(obs, pred, na.rm = TRUE)
```

Arguments

obs	Numeric observation vector.
pred	Numeric prediction vector paired with obs.
na.rm	Logical; remove incomplete pairs?

Details

$$r^2 = \frac{\left[\sum_{i=1}^{n}(obs_i - \bar{obs})(pred_i - \bar{pred})\right]^2}{\sum_{i=1}^{n}(obs_i - \bar{obs})^2 \sum_{i=1}^{n}(pred_i - \bar{pred})^2}.$$

It ranges from zero to one and summarizes the strength, but not the sign, of linear association. It describes the dispersion of predictions and observations around their fitted linear relationship rather than their departure from the 1:1 line. Consequently, r2() is insensitive to additive bias and proportional scaling: a value of one can occur even when predictions are systematically biased or have a different scale from the observations. It should therefore not be interpreted as a general measure of predictive agreement or accuracy.

Do not confuse lowercase `r2()` with `nse()`, `mec()`, or uppercase `R2()`, which are model-efficiency statistics and are sensitive to departures from the line of equality. It returns `NA` with a warning when fewer than two valid pairs remain or either vector has zero variance.

Value

One numeric value.

References

Willmott, C. J. (1984). On the evaluation of model performance in physical geography. In G. L. Gaile and C. J. Willmott (Eds.), *Spatial Statistics and Models* (pp. 443-460). D. Reidel.

Legates, D. R. and McCabe, G. J. (1999). Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation. *Water Resources Research*, 35(1), 233-241. doi:10.1029/1998WR900018

Examples

```
r2(1:3, c(1, 3, 2))
```

---

rae *Relative absolute error*

Description

Relative absolute error (RAE) compares total absolute error with total absolute error from predicting the observed mean.

Usage

```
rae(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\mathrm{RAE} = \frac{\sum_{i=1}^{n} |obs_i - pred_i|}{\sum_{i=1}^{n} |obs_i - \bar{obs}|}.$$

RAE is non-negative and zero is ideal. One equals the observed-mean absolute-error benchmark; values above one are worse. It returns `NA` with a warning for constant observations, whose benchmark denominator is zero. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Hyndman and Koehler (2006). See `mdae()`.

See Also

Other prediction metrics: `mape()`, `mdae()`, `mpe()`, `msle()`, `rer()`, `rmsle()`, `rpd()`, `rpiq()`, `rrmse()`, `sep()`, `smape()`, `willmott_d()`

---

rer *Range-to-RMSE ratio*

Description

Range-to-RMSE ratio (RER) scales RMSE by the observed range.

Usage

```
rer(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\mathrm{RER} = \frac{\max_i(obs_i) - \min_i(obs_i)}{\sqrt{n^{-1}\sum_{i=1}^{n}(obs_i - pred_i)^2}}.$$

RER is non-negative and larger values indicate smaller error relative to the observed range. It is sensitive to extreme observations. For perfect predictions with a nonzero range it returns `Inf`; it returns `NA` with a warning for the indeterminate zero-over-zero case. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Bellon-Maurel et al. (2010). See `rpd()`.

See Also

Other prediction metrics: `mape()`, `mdae()`, `mpe()`, `msle()`, `rae()`, `rmsle()`, `rpd()`, `rpiq()`, `rrmse()`, `sep()`, `smape()`, `willmott_d()`

`rmse` *Root mean squared error*

Description

Root mean squared error (RMSE) is the square root of mean squared error.

Usage

```
rmse(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(obs_i - pred_i)^2}$$

RMSE is non-negative, has the same units as the response variable, and is zero for perfect predictions. Smaller values indicate better agreement. Squaring means that a small number of large errors can disproportionately increase RMSE. Therefore, when errors are skewed or asymmetric, interpret RMSE together with `mae()` and inspect the error distribution rather than relying on RMSE alone. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Legates, D. R. and McCabe, G. J. (1999). Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation. *Water Resources Research*, 35(1), 233-241. doi:10.1029/1998WR900018

Armstrong, J. S. (2001). Evaluating forecasting methods. In J. S. Armstrong (Ed.), *Principles of Forecasting* (pp. 443-472). Springer. doi:10.1007/978-0-306-47630-3_20

Willmott, C. J. and Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30, 79-82. doi:10.3354/cr030079

Hodson, T. O. (2022). Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development*, 15, 5481-5487. doi:10.5194/gmd-15-5481-2022

Examples

```
rmse(1:3, c(1, 3, 2))
```

`rmsle` *Root mean squared logarithmic error*

Description

Root mean squared logarithmic error (RMSLE) is the square root of `msle()`.

Usage

```
rmsle(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\text{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[\log(1+obs_i) - \log(1+pred_i)]^2}.$$

RMSLE is non-negative and zero is ideal. It has the same non-negative input requirement and log1p convention as `msle()`, returning `NA` with a warning when either input contains a negative value. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Hodson (2022). See `msle()`.

See Also

Other prediction metrics: `mape()`, `mdae()`, `mpe()`, `msle()`, `rae()`, `rer()`, `rpd()`, `rpiq()`, `rrmse()`, `sep()`, `smape()`, `willmott_d()`

rpd *Ratio of performance to deviation*

Description

Ratio of performance to deviation (RPD) scales RMSE by the sample standard deviation of observations.

Usage

```
rpd(obs, pred, na.rm = TRUE)
```

Arguments

obs	Numeric observation vector.
pred	Numeric prediction vector paired with obs.
na.rm	Logical; remove incomplete pairs?

Details

$$\mathrm{RPD} = \frac{\sqrt{(n-1)^{-1}\sum_{i=1}^{n}(obs_i - \bar{obs})^2}}{\sqrt{n^{-1}\sum_{i=1}^{n}(obs_i - pred_i)^2}}.$$

RPD is non-negative and larger values indicate lower error relative to observed variation. For perfect predictions with nonzero observed variation it returns Inf, the mathematically defined ratio with zero RMSE, without a warning. It returns NA with a warning for fewer than two retained pairs or for the indeterminate zero-over-zero case. Missing-value handling follows bias().

Value

One numeric value.

References

Bellon-Maurel, V., Fernandez-Ahumada, E., Palagos, B., Roger, J.-M., and McBratney, A. (2010). Critical review of chemometric indicators commonly used for assessing the quality of the prediction of soil attributes by NIR spectroscopy. *Trends in Analytical Chemistry*, 29, 1073-1081. doi:10.1016/j.trac.2010.05.006

See Also

Other prediction metrics: mape(), mdae(), mpe(), msle(), rae(), rer(), rmsle(), rpiq(), rrmse(), sep(), smape(), willmott_d()

`rpiq` *Ratio of performance to interquartile distance*

Description

Ratio of performance to interquartile distance (RPIQ) scales RMSE by the interquartile range of observations.

Usage

```
rpiq(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\mathrm{RPIQ} = \frac{Q_{0.75}(obs) - Q_{0.25}(obs)}{\sqrt{n^{-1}\sum_{i=1}^{n}(obs_i - pred_i)^2}}.$$

RPIQ is non-negative and larger values indicate lower error relative to the middle 50 percent of observed values. It is less influenced by extremes than `rpd()`. R uses default type-7 quartiles. For perfect predictions with a nonzero interquartile range it returns `Inf`; it returns `NA` with a warning for the indeterminate zero-over-zero case. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Bellon-Maurel et al. (2010). See `rpd()`.

See Also

Other prediction metrics: `mape()`, `mdae()`, `mpe()`, `msle()`, `rae()`, `rer()`, `rmsle()`, `rpd()`, `rrmse()`, `sep()`, `smape()`, `willmott_d()`

`rrmse` *Relative root mean squared error*

Description

Relative root mean squared error (RRMSE) expresses RMSE as a percentage of the absolute observed mean.

Usage

```
rrmse(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\mathrm{RRMSE} = 100\,\frac{\sqrt{n^{-1}\sum_{i=1}^{n}(obs_i - pred_i)^2}}{|\bar{obs}|}.$$

RRMSE is non-negative and zero is ideal. It returns `NA` with a warning for a zero observed mean and is unstable when that mean is near zero. This mean-normalized convention differs from the standard-deviation normalization in `nrmse()`. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Willmott, C. J., Ackleson, S. G., Davis, R. E., Feddema, J. J., Klink, K. M., Legates, D. R., O'Donnell, J., and Rowe, C. M. (1985). Statistics for the evaluation and comparison of models. *Journal of Geophysical Research*, 90, 8995-9005. doi:10.1029/JC090iC05p08995

See Also

Other prediction metrics: `mape()`, `mdae()`, `mpe()`, `msle()`, `rae()`, `rer()`, `rmsle()`, `rpd()`, `rpiq()`, `sep()`, `smape()`, `willmott_d()`

`sd_ratio` *Prediction-to-observation standard deviation ratio*

Description

The standard deviation ratio compares the sample standard deviation of predictions with that of observations.

Usage

```
sd_ratio(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\text{SD ratio} = \sqrt{\frac{\sum_{i=1}^{n}(pred_i - \bar{pred})^2}{\sum_{i=1}^{n}(obs_i - \bar{obs})^2}}.$$

The ratio is non-negative and one indicates equal variability. Values below one indicate that predictions have less variability than the observations; values above one indicate that predictions have more variability than the observations. It assesses variability, not mean bias or association, and returns `NA` with a warning when fewer than two valid pairs remain or the observations have zero variance. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Taylor, K. E. (2001). Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research*, 106, 7183-7192. doi:10.1029/2000JD900719

Examples

```
sd_ratio(1:3, c(1, 3, 2))
```

`sep` *Standard error of prediction*

Description

Standard error of prediction (SEP) is the sample standard deviation of prediction errors after removing their mean error (ME).

Usage

```
sep(obs, pred, na.rm = TRUE)
```

Arguments

`obs`	Numeric observation vector.
`pred`	Numeric prediction vector paired with `obs`.
`na.rm`	Logical; remove incomplete pairs?

Details

$$\mathrm{SEP} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left[(obs_i - pred_i) - \frac{1}{n}\sum_{j=1}^{n}(obs_j - pred_j)\right]^2}.$$

SEP has response units, is non-negative, and zero is ideal. Unlike RMSE, it removes constant bias. It returns `NA` with a warning when fewer than two retained pairs remain. Missing-value handling follows `bias()`.

Value

One numeric value.

References

Bellon-Maurel et al. (2010). See `rpd()`.

See Also

Other prediction metrics: `mape()`, `mdae()`, `mpe()`, `msle()`, `rae()`, `rer()`, `rmsle()`, `rpd()`, `rpiq()`, `rrmse()`, `smape()`, `willmott_d()`

smape *Symmetric mean absolute percentage error*

Description

Symmetric mean absolute percentage error (sMAPE) scales absolute error by absolute observation and prediction sizes.

Usage

```
smape(obs, pred, na.rm = TRUE)
```

Arguments

obs	Numeric observation vector.
pred	Numeric prediction vector paired with obs.
na.rm	Logical; remove incomplete pairs?

Details

$$\text{sMAPE} = \frac{100}{n} \sum_{i=1}^{n} \frac{2|obs_i - pred_i|}{|obs_i| + |pred_i|}.$$

sMAPE ranges from zero to 200 percent; zero is ideal. A pair of zeros contributes zero. Missing-value handling follows bias().

Value

One numeric value.

References

Hyndman and Koehler (2006). See mdae().

See Also

Other prediction metrics: mape(), mdae(), mpe(), msle(), rae(), rer(), rmsle(), rpd(), rpiq(), rrmse(), sep(), willmott_d()

`uncertainty_metrics` *Summarize prediction-interval validation*

Description

Calculates a set of complementary diagnostics for evaluating central prediction intervals. The function combines empirical coverage, departure from nominal coverage, interval width, and the interval score in a single one-row summary.

Usage

```
uncertainty_metrics(
  obs,
  lower = NULL,
  upper = NULL,
  level = 0.95,
  na.rm = TRUE,
  pred = NULL,
  predictive_sd = NULL,
  distribution = NULL
)
```

Arguments

`obs`	Numeric observation vector.
`lower`, `upper`	Numeric vectors containing the lower and upper prediction-interval bounds. In explicit-bound mode, both must be supplied and have the same length as `obs`.
`level`	Nominal central prediction-interval coverage probability, strictly between zero and one. The default is `0.95`.
`na.rm`	Logical; remove incomplete observation/interval combinations? The default is `TRUE`.
`pred`, `predictive_sd`	Optional numeric vectors of predictive means and predictive standard deviations, supplied together and of the same length as `obs`. Assumes normal predictive distributions. Non-missing predictive standard deviations must be finite and strictly positive.
`distribution`	Optional numeric matrix or data frame of equally weighted predictive samples: one row per observation and one column per predictive draw. Supply this instead of bounds or predictive means and standard deviations. At least one draw is required; infinite values are not allowed.

Details

Supply `lower` and `upper`, predictive mean and standard deviation (`pred` and `predictive_sd`), or predictive samples (`distribution`). For a nominal interval level p, a well-calibrated uncertainty model should have empirical prediction interval coverage probability (PICP) close to p.

The returned diagnostics describe complementary aspects of prediction- interval performance:

- picp is the proportion of observations contained within the prediction intervals;
- picp_error is empirical minus nominal coverage. Values below zero indicate under-coverage, whereas values above zero indicate over-coverage;
- interval_width is the mean width of the prediction intervals and summarizes their sharpness. Smaller values indicate narrower intervals, but are desirable only when calibration remains adequate;
- interval_score is a proper scoring rule that rewards narrow intervals while penalizing observations falling below or above them. Smaller values indicate better performance when comparing predictions on the same response scale and at the same nominal interval level.

These quantities should be interpreted together. Coverage alone does not reward sharp predictions, while interval width alone does not assess whether the intervals contain observations at the stated frequency. The interval score combines both aspects in a single statistic.

This function summarizes prediction intervals at one level. For calibration across multiple interval levels, use gg_coverage() and accuracy_plot_metrics(). For predictive quantiles, use qcp() and gg_qcp(). For complete predictive distributions, use pit(), gg_pit(), crps(), or crps_decomposition().

All four returned statistics use the same complete observation/lower/upper triplets. This differs intentionally from standalone interval_width(), which is observation-independent and can use interval bounds where obs is missing.

Value

A one-row base data frame containing:

picp Empirical prediction interval coverage probability.

picp_error Difference between empirical and nominal coverage, calculated as picp - level.

interval_width Mean prediction-interval width.

interval_score Mean interval score at the specified nominal coverage level.

Input representations

Supply exactly one of explicit lower and upper bounds, predictive mean and standard deviation (pred and predictive_sd), or predictive samples (distribution). Normal inputs generate central intervals using normal quantiles. Samples generate equal-tailed intervals using stats::quantile() with type = 7, at probabilities (1 - level) / 2 and (1 + level) / 2. With na.rm = TRUE, a case is removed if its observation or any supplied predictive value is missing; individual missing draws are not discarded within a case. With na.rm = FALSE, incomplete inputs give NA. No complete cases also gives NA. Standalone interval_width() ignores missing obs.

References

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102, 359-378. doi:10.1198/016214506000001437

Schmidinger, J. and Heuvelink, G. B. M. (2023). Validation of uncertainty predictions in digital soil mapping. *Geoderma*, 437, 116585. doi:10.1016/j.geoderma.2023.116585

See Also

picp(), coverage_error(), interval_width(), interval_score(), gg_coverage(), accuracy_plot_metrics(), qcp(), pit(), crps()

Examples

```
obs <- c(1, 2, 3, 4, 5)
lower <- c(0, 1, 2, 3, 4)
upper <- c(2, 3, 4, 5, 6)

uncertainty_metrics(
  obs,
  lower = lower,
  upper = upper,
  level = 0.80
)
uncertainty_metrics(obs, pred = obs, predictive_sd = rep(1, 5), level = 0.8)
uncertainty_metrics(obs, distribution = cbind(obs - 1, obs, obs + 1))
```

---

willmott_d *Willmott's index of agreement*

Description

Willmott's original index of agreement, d, compares squared error with a potential-error denominator based on the observed mean.

Usage

```
willmott_d(obs, pred, na.rm = TRUE)
```

Arguments

obs	Numeric observation vector.
pred	Numeric prediction vector paired with obs.
na.rm	Logical; remove incomplete pairs?

Details

$$d = 1 - \frac{\sum_{i=1}^{n}(obs_i - pred_i)^2}{\sum_{i=1}^{n}(|pred_i - \bar{obs}| + |obs_i - \bar{obs}|)^2}.$$

For finite inputs, d ranges from zero to one and one is ideal. The index can be strongly influenced by large errors. It returns NA with a warning when its denominator is zero, including identical constant observations and predictions. Missing-value handling follows bias().

Value

One numeric value.

References

Willmott et al. (1985). See `rrmse()`.

See Also

Other prediction metrics: `mape()`, `mdae()`, `mpe()`, `msle()`, `rae()`, `rer()`, `rmsle()`, `rpd()`, `rpiq()`, `rrmse()`, `sep()`, `smape()`

Index