Package 'metrica'

July 22, 2025

Title Prediction Performance Metrics

Version 2.1.0

Date 2024-06-30

Description A compilation of more than 80 functions designed to quantitatively and visually evaluate prediction performance of regression (continuous variables) and classification (categorical variables) of point-forecast models (e.g. APSIM, DSSAT, DNDC, supervised Machine Learning). For regression, it includes functions to generate plots (scatter, tiles, density, & Bland-Altman plot), and to estimate error metrics (e.g. MBE, MAE, RMSE), error decomposition (e.g. lack of accuracy-precision), model efficiency (e.g. NSE, E1, KGE), indices of agreement (e.g. d, RAC), goodness of fit (e.g. r, R2), adjusted correlation coefficients (e.g. CCC, dcorr), symmetric regression coefficients (intercept, slope), and mean absolute scaled error (MASE) for time series predictions. For classification (binomial and multinomial), it offers functions to generate and plot confusion matrices, and to estimate performance metrics such as accuracy, precision, recall, specificity, F-score, Cohen's Kappa, G-mean, and many more. For more details visit the vignettes <https://adriancorrendo.github.io/metrica/>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Depends R (>= 3.6.0)

Suggests purrr, knitr, rmarkdown, apsimx, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports stats, ggplot2, dplyr, rlang, tidyr, utils, DBI, RSQLite, ggpp, minerva, energy

VignetteBuilder knitr

URL https://adriancorrendo.github.io/metrica/

BugReports https://github.com/adriancorrendo/metrica/issues

NeedsCompilation no

Author Adrian A. Correndo [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-4172-289X>),
Luiz H. Moro Rosso [aut] (ORCID:
<https://orcid.org/0000-0002-8642-911X>),
Rai Schwalbert [aut] (ORCID: <https://orcid.org/0000-0001-8488-7507>),
Carlos Hernandez [aut] (ORCID: <https://orcid.org/0000-0001-5171-2516>),
Leonardo M. Bastos [aut] (ORCID:
<https://orcid.org/0000-0001-8958-6527>),
Luciana Nieto [aut] (ORCID: <https://orcid.org/0000-0002-7172-0799>),
Dean Holzworth [aut],
Ignacio A. Ciampitti [aut] (ORCID:
<https://orcid.org/0000-0001-9619-5129>)

Maintainer Adrian A. Correndo <acorrend@uoguelph.ca>

Repository CRAN

Date/Publication 2024-06-30 14:20:02 UTC

Contents

AC *Ji and Gallo's Agreement Coefficient (AC)*

Description

It estimates the agreement coefficient suggested by Ji & Gallo (2006) for a continuous predicted-observed dataset.

Usage

```
AC(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The Ji and Gallo's AC measures general agreement, including both accuracy and precision. It is normalized, dimensionless, positively bounded (-infinity;1), and symmetric. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Ji & Gallo (2006). An agreement coefficient for image comparison. *Photogramm. Eng. Remote Sensing 7, 823–833* doi:10.14358/PERS.72.7.823

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
df <- data.frame(obs = X, pred = Y)
AC(df, obs = X, pred = Y)
```

`accuracy` *Accuracy*

Description

It estimates the accuracy for a nominal/categorical predicted-observed dataset.

Usage

```
accuracy(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (character | factor).
`pred`	Vector with predicted values (character | factor).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list (default).
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

Accuracy is the simplest and most popular classification metric in literature. It refers to a measure of the degree to which the predictions of a model matches the reality being modeled. The classification accuracy is calculated as the ratio between the number of correctly classified objects with respect to the total number of cases.

It is bounded between 0 and 1. The closer to 1 the better. Values towards zero indicate low accuracy of predictions. It can be also expressed as percentage if multiplied by 100. It is estimated at a global level (not at the class level).

Accuracy presents limitations to address classification quality under unbalanced classes, and it is not able to distinguish among misclassification distributions. For those cases, it is advised to apply other metrics such as balanced accuracy (baccu), F-score (fscore), Matthews Correlation Coefficient (mcc), or Cohen's Kappa Coefficient (cohen_kappa).

Accuracy is directly related to the error_rate, since accuracy = 1 – error_rate.

For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Sammut & Webb (2017). Accuracy. *In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA.* doi:10.1007/9781489976871_3

See Also

`balacc` `fscore` `mcc` `khat`

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100,
replace = TRUE), predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100,
replace = TRUE), predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE) )

# Get accuracy estimate for two-class case
accuracy(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get accuracy estimate for multi-class case
accuracy(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)
```

---

agf *Adjusted F-score*

Description

It estimates the Adjusted F-score for a nominal/categorical predicted-observed dataset.

Usage

```
agf(
  data = NULL,
  obs,
  pred,
  pos_level = 2,
  atom = FALSE,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (character | factor).
`pred`	Vector with predicted values (character | factor).
`pos_level`	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are `(Negative | Positive)`, `(0 | 1)`, `(FALSE | TRUE)`. Default : 2.
`atom`	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE. When dataset is "binomial" atom does not apply.
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The Adjusted F-score (or Adjusted F-measure) is an improvement over the F1-score especially when the data classes are imbalanced. This metric more properly accounts for the different misclassification costs across classes. It weights more the sensitivity (recall) metric than precision and gives strength to the false negative values. This index accounts for all elements of the original confusion matrix and provides more weight to patterns correctly classified in the minority class (positive).

It is bounded between 0 and 1. The closer to 1 the better. Values towards zero indicate low performance. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Maratea, A., Petrosino, A., Manzo, M. (2014). Adjusted-F measure and kernel scaling for imbalanced data learning. *Inf. Sci. 257: 331-341.* doi:10.1016/j.ins.2013.04.016

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100, replace = TRUE),
predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE)    )

# Get F-score estimate for two-class case
```

```
agf(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get F-score estimate for each class for the multi-class case
agf(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get F-score estimate for the multi-class case at a global level
agf(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)
```

AUC_roc *Area Under the ROC Curve*

Description

The AUC estimates the area under the receiver operator curve (ROC) for a nominal/categorical predicted-observed dataset using the Mann-Whitney U-statistic.

Usage

```
AUC_roc(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list (default).
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The AUC tests whether positives are ranked higher than negatives. It is simply the area under the ROC curve.

The AUC estimation of this function follows the procedure described by Hand & Till (2001). The AUC_roc estimated following the trapezoid approach is equivalent to the average between recall and specificity (Powers, 2011), which is equivalent to the balanced accuracy (balacc):

$AUC_roc = \frac{(recall - FPR + 1)}{2} = \frac{recall + specificity}{2} = 1 - \frac{FPR + FNR}{2}$

Interpretation: the AUC is equivalent to the probability that a randomly case from a given class (positive for binary) will have a smaller estimated probability of belonging to another class (negative for binary) compared to a randomly chosen member of the another class.

Values: the AUC is bounded between 0 and 1. The closer to 1 the better. Values close to 0 indicate inaccurate predictions. An AUC = 0.5 suggests no discrimination ability between classes; 0.7 < AUC < 0.8 is considered acceptable, 0.8 < AUC < 0.5 is considered excellent, and AUC > 0.9 is outstanding (Mandrekar, 2010).

For the multiclass cases, the AUC is equivalent to the average of AUC of each class (Hand & Till, 2001).

Finally, the AUC is directly related to the Gini-index (a.k.a. G1) since Gini + 1 = 2*AUC. (Hand & Till, 2001).

For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Hanley, J.A., McNeil, J.A. (2017). The meaning and use of the area under a receiver operating characteristic (ROC) curve. _ Radiology 143(1): 29-36_ doi:10.1148/radiology.143.1.7063747

Hand, D.J., Till, R.J. (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems. _ Machine Learning 45: 171-186_ doi:10.1023/A:1010920819831

Mandrekar, J.N. (2010). Receiver operating characteristic curve in diagnostic test assessment. _ J. Thoracic Oncology 5(9): 1315-1316_ doi:10.1097/JTO.0b013e3181ec173d

Powers, D.M.W. (2011). Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies 2(1): 37–63.* doi:10.48550/arXiv.2010.16061

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100,
replace = TRUE), predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100,
replace = TRUE), predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE) )

# Get AUC estimate for two-class case
AUC_roc(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get AUC estimate for multi-class case
AUC_roc(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)
```

`B0_sma` *Intercept of standardized major axis regression (SMA).*

Description

It calculates the intercept (B0) for the bivariate linear relationship between predicted and observed values following the SMA regression.

Usage

```
B0_sma(data = NULL, obs, pred, orientation = "PO", tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
orientation	Argument of class string specifying the axis orientation, PO for predicted vs observed, and OP for observed vs predicted. Default is orientation = "PO".
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na_rm = TRUE.

Details

SMA is a symmetric linear regression (invariant results/interpretation to axis orientation) recommended to describe the bivariate scatter instead of OLS regression (classic linear model, which results vary with the axis orientation). For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

References

Warton et al. (2006). Bivariate line-fitting methods for allometry. *Biol. Rev. Camb. Philos. Soc. 81, 259–291.* doi:10.1002/15214036(200203)44:2<161::AIDBIMJ161>3.0.CO;2N

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
B0_sma(obs = X, pred = Y)
```

B1_sma *Slope of standardized major axis regression (SMA).*

Description

It calculates the slope (B1) for the bivariate linear relationship between predicted and observed values following the SMA regression.

Usage

```
B1_sma(data = NULL, obs, pred, tidy = FALSE, orientation = "PO", na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
orientation	Argument of class string specifying the axis orientation, PO for predicted vs observed, and OP for observed vs predicted. Default is orientation = "PO".
na.rm	Logic argument to remove rows with missing values (NA). Default is na_rm = TRUE.

Details

SMA is a symmetric linear regression (invariant results/interpretation to axis orientation) recommended to describe the bivariate scatter instead of OLS regression (classic linear model, which results vary with the axis orientation). For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Warton et al. (2006). Bivariate line-fitting methods for allometry. *Biol. Rev. Camb. Philos. Soc. 81, 259–291.* doi:10.1002/15214036(200203)44:2<161::AIDBIMJ161>3.0.CO;2N

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
B1_sma(obs = X, pred = Y)
```

balacc *Balanced Accuracy*

Description

It estimates the balanced accuracy for a nominal/categorical predicted-observed dataset.

Usage

```
balacc(data = NULL, obs, pred, pos_level = 2, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The balanced accuracy is the average between recall and specificity. It is particularly useful when the number of observation belonging to each class is despair or imbalanced, and when especial attention is given to the negative cases. It is bounded between 0 and 1. Values towards zero indicate low performance. The closer to 1 the better.

For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

References

García, V., Mollineda, R.A., Sánchez, J.S. (2009). Index of Balanced Accuracy: A Performance Measure for Skewed Class Distributions. *In: Araujo, H., Mendonça, A.M., Pinho, A.J., Torres, M.I. (eds) Pattern Recognition and Image Analysis. IbPRIA 2009. Lecture Notes in Computer Science, vol 5524. Springer-Verlag Berlin Heidelberg.* doi:10.1007/9783642021725_57

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100, replace = TRUE),
predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE)    )

# Get balanced accuracy estimate for two-class case
balacc(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get balanced accuracy estimate for the multi-class case
balacc(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)
```

---

barley *Barley grain number*

Description

This example dataset is a set of APSIM simulations of barley grain number (x1000 grains per squared meter), which presents high accuracy but medium precision. The experimental trials come from 11 site-years in 2 countries (Australia, and New Zealand). These data correspond to the latest, up-to-date, documentation and validation of version number 2020.03.27.4956. Data available at: doi:10.7910/DVN/EJS4M0. Further details can be found at the official APSIM Next Generation website: https://APSIMnextgeneration.netlify.app/modeldocumentation/

Usage

```
barley
```

Format

This data frame has 69 rows and the following 2 columns:

pred predicted values

obs observed values

Source

https://github.com/adriancorrendo/metrica

`bland_altman_plot` *Bland-Altman plot*

Description

It creates a scatter plot of the difference between observed and predicted values (obs-pred) vs. observed values.

Usage

```
bland_altman_plot(
  data = NULL,
  obs,
  pred,
  shape_type = NULL,
  shape_size = NULL,
  shape_color = NULL,
  shape_fill = NULL,
  zeroline_type = NULL,
  zeroline_size = NULL,
  zeroline_color = NULL,
  limitsline_type = NULL,
  limitsline_size = NULL,
  limitsline_color = NULL,
  na.rm = TRUE
)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`shape_type`	number indicating the shape type for the data points.
`shape_size`	number indicating the shape size for the data points.
`shape_color`	string indicating the shape color for the data points.
`shape_fill`	string indicating the shape fill for the data points.
`zeroline_type`	string or integer indicating the zero line-type.
`zeroline_size`	number indicating the zero line size.
`zeroline_color`	string indicating the zero line color.
`limitsline_type`	string or integer indicating the limits (+/- 1.96*SD) line-type.
`limitsline_size`	number indicating the limits (+/- 1.96*SD) line size.
`limitsline_color`	string indicating the limits (+/- 1.96*SD) line color.
`na.rm`	Logic argument to remove rows with missing values

Details

For more details, see online-documentation

Value

an object of class `ggplot`.

References

Bland & Altman (1986). Statistical methods for assessing agreement between two methods of clinical measurement *The Lancet 327(8476), 307-310* doi:10.1016/S01406736(86)908378

See Also

`ggplot`,`geom_point`,`aes`

Examples

```
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 10)
bland_altman_plot(obs = X, pred = Y)
```

---

bmi *Bookmaker Informedness*

Description

It estimates the Bookmaker Informedness (a.k.a. Youden's J-index) for a nominal/categorical predicted-observed dataset.

`jindex` estimates the Youden's J statistic or Youden's J Index (equivalent to Bookmaker Informedness `bmi`)

Usage

```
bmi(
  data = NULL,
  obs,
  pred,
  pos_level = 2,
  atom = FALSE,
  tidy = FALSE,
  na.rm = TRUE
)

jindex(
  data = NULL,
```

```
  obs,
  pred,
  pos_level = 2,
  atom = FALSE,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.
atom	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE. When dataset is "binomial" atom does not apply.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The Bookmaker Informedness (or Youden's J index) it is a suitable metric when the number of cases for each class is uneven.

The general formula applied to both binary and multiclass cases is:

$bmi = recall + specificity - 1$

It is bounded between 0 and 1. The closer to 1 the better. Values towards zero indicate low performance. For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

References

Youden, W.J. (1950). Index for rating diagnostic tests. . *Cancer 3: 32-35.* doi:10.1002/1097-0142(1950)3:1<32::AIDCNCR2820030106>3.0.CO;23

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100, replace = TRUE),
predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE)    )

# Get Informedness estimate for two-class case
bmi(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get Informedness estimate for each class for the multi-class case
bmi(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE, atom = TRUE)

# Get Informedness estimate for the multi-class case at a global level
bmi(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)
```

CCC *Concordance correlation coefficient (CCC)*

Description

It estimates the Concordance Correlation Coefficient (CCC) for a continuous predicted-observed dataset.

Usage

```
CCC(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The CCC it is a normalized coefficient that tests general agreement. It presents both precision (r) and accuracy (Xa) components. It is positively bounded to 1. The closer to 1 the better. Values towards zero indicate low correlation between observations and predictions. Negative values would indicate a negative relationship between predicted and observed. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Lin (1989). A concordance correlation coefficient to evaluate reproducibility. *Biometrics 45 (1), 255–268.* doi:10.2307/2532051

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
CCC(obs = X, pred = Y)
```

---

chickpea *Chickpea dry mass*

Description

This example dataset is a set of APSIM simulations of chickpea aboveground dry mass (kg per hectare), which exhibits low accuracy and medium-low precision as it represents a model still under development. The experimental trials come from 7 site-years in 3 countries (Australia, India, and New Zealand). These data correspond to the latest, up-to-date, documentation and validation of version number 2020.03.27.4956. Data available at: doi:10.7910/DVN/EJS4M0. Further details can be found at the official APSIM Next Generation website: https://APSIMnextgeneration.netlify.app/modeldocumentation/

Usage

```
chickpea
```

Format

This data frame has 39 rows and the following 2 columns:

pred predicted values

obs observed values

Source

https://github.com/adriancorrendo/metrica

confusion_matrix *Confusion Matrix*

Description

It creates a confusion matrix table or plot displaying the agreement between the observed and the predicted classes by the model.

Usage

```
confusion_matrix(
  data = NULL,
  obs,
  pred,
  plot = FALSE,
  unit = "count",
  colors = c(low = NULL, high = NULL),
  print_metrics = FALSE,
  metrics_list = c("accuracy", "precision", "recall"),
  position_metrics = "top",
  na.rm = TRUE
)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character or factor).
pred	Vector with predicted values (character or factor).
plot	Logical operator (TRUE/FALSE) that controls the output as a data.frame (plot = FALSE) or as a plot of type ggplot (plot = TRUE), Default: FALSE
unit	String (text) indicating the type of unit ("count" or "proportion") to show in the confusion matrix, Default: 'count'
colors	Vector or list with two colors indicating how to paint the gradient between "low" and "high", Default: c(low = NULL, high = NULL) uses the standard blue gradient of ggplot2.
print_metrics	boolean TRUE/FALSE to embed metrics in the plot. Default is FALSE.
metrics_list	vector or list of selected metrics to print on the plot. Default: c("accuracy", "precision", "recall").
position_metrics	string specifying the position to print the performance metrics_list. Options are "top" (as a subtitle) or "bottom" (as a caption). Default: "bottom".
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

A confusion matrix is a method for summarizing the predictive performance of a classification algorithm. It is particularly useful if you have an unbalanced number of observations belonging to each class or if you have a multinomial dataset (more than two classes in your dataset. A confusion matrix can give you a good hint about the types of errors that your model is making. See online-documentation

Value

An object of class `data.frame` when plot = FALSE, or of type `ggplot` when plot = TRUE.

References

Ting K.M. (2017). Confusion Matrix. *In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA.* doi:10.1007/9781489976871_50

See Also

`eval_tidy`, `defusing-advanced` `select`

Examples

```
set.seed(183)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100,
replace = TRUE), predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE))

# Plot two-class confusion matrix
confusion_matrix(data = binomial_case, obs = labels, pred = predictions,
plot = TRUE, colors = c(low="pink" , high="steelblue"), unit = "count")

# Plot multi-class confusion matrix
confusion_matrix(data = multinomial_case, obs = labels, pred = predictions,
plot = TRUE, colors = c(low="#f9dbbd" , high="#735d78"), unit = "count")
```

`csi` *Critical Success Index | Jaccard's Index*

Description

It estimates the Critical Success Index (a.k.a. threat score, Jaccard Index) for a nominal/categorical predicted-observed dataset.

`jaccardindex` estimates the Jaccard similarity coefficient or Jaccard's Index (equivalent to the Critical Success Index `csi`).

Usage

```
csi(
  data = NULL,
  obs,
  pred,
  pos_level = 2,
  atom = FALSE,
  tidy = FALSE,
  na.rm = TRUE
)

jaccardindex(
  data = NULL,
  obs,
  pred,
  pos_level = 2,
  atom = FALSE,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (character | factor).
`pred`	Vector with predicted values (character | factor).
`pos_level`	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are `(Negative | Positive)`, `(0 | 1)`, `(FALSE | TRUE)`. Default : 2.
`atom`	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE. When dataset is "binomial" atom does not apply.
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The `csi` is also known as the threat score or the Jaccard Index. It is a metric especially useful for binary classification tasks, representing the proportion of true positive (TP) cases with respect to the sum of predicted positive (PP = TP + FP) and true negative (TN) cases.

$csi = \frac{TP}{TP+TN+FP}$

It is bounded between 0 and 1. The closer to 1 the better the classification performance, while zero represents the worst result.

It has been extensively used in meteorology (NOOA) as a verification measure of categorical forecast performance equal to the total number of correct event forecast (hits = TP) divided by the total number of event forecasts plus the number of misses (hits + false alarms + misses = TP + FP + TN). However, the csi has been criticized for not representing an unbiased measure of forecast skill (Schaefer, 1990).

For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

NOOA. Forecast Verification Glossary. *Space Weather Prediction Center, NOOA*. https://www.swpc.noaa.gov/sites/default/files/images/u30/Forecast%20Verification%20Glossary.pdf

Schaefer, J.T. (1990). The critical success index as an indicator of warning skill. *Weather and Forecasting 5(4): 570-575*. doi:10.1175/15200434(1990)005<0570:TCSIAA>2.0.CO;2

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))
# Get csi estimate for two-class case
csi(data = binomial_case, obs = labels, pred = predictions)
```

---

d *Willmott's Index of Agreement (d)*

Description

It estimates the Willmott's index of agreement (d) for a continuous predicted-observed dataset.

Usage

```
d(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The d index it is a normalized, dimensionless metric that tests general agreement. It measures both accuracy and precision using squared residuals. It is bounded between 0 and 1. The disadvantage is that d is an asymmetric index, that is, dependent to what is orientation of predicted and observed values. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Willmott (1981). On the validation of models. *Phys. Geogr. 2, 184–194.* doi:10.1080/02723646.1981.10642213

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
d(obs = X, pred = Y)
```

---

d1 *Modified Index of Agreement (d1).*

Description

It estimates the modified index of agreement (d1) using absolute differences following Willmott et al. (1985).

Usage

```
d1(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

Similar to d, the d1 index it is a normalized, dimensionless metric that tests general agreement. The difference with d, is that d1 uses absolute residuals instead of squared residuals. It is bounded between 0 and 1. The disadvantage is that d is an asymmetric index, that is, dependent to the orientation of predicted and observed values. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Willmott et al. (1985). Statistics for the evaluation and comparison of models. *J. Geophys. Res. 90, 8995.* doi:10.1029/jc090ic05p08995

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
d1(obs = X, pred = Y)
```

---

d1r *Refined Index of Agreement (d1).*

Description

It estimates the refined index of agreement (d1r) following Willmott et al. (2012).

Usage

```
d1r(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

Similar to d, and d1, the d1r index it is a normalized, dimensionless metric that tests general agreement. The difference is that d1r modifies the denominator of the formula (potential error), normalizing the mean absolute error (numerator) by two-times the mean absolute deviation of observed values. It is bounded between 0 and 1. The disadvantage is that d1r is an asymmetric index, that is, dependent to the orientation of predicted and observed values. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Willmott et al. (2012). A refined index of model performance. *Int. J. Climatol. 32, 2088–2094.* doi:10.1002/joc.2419

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
d1r(obs = X, pred = Y)
```

---

dcorr *Distance Correlation*

Description

It estimates the Distance Correlation coefficient (dcorr) for a continuous predicted-observed dataset.

Usage

```
dcorr(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list (default).
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The **dcorr** function is a wrapper for the `dcor` function from the **energy**-package. See Rizzo & Szekely (2022). The distance correlation (dcorr) coefficient is a novel measure of dependence between random vectors introduced by Szekely et al. (2007).

The dcorr is characterized for being symmetric, which is relevant for the predicted-observed case (PO).

For all distributions with finite first moments, distance correlation $\mathcal{R}$ generalizes the idea of correlation in two fundamental ways:

(1) $\mathcal{R}(P, O)$ is defined for P and O in arbitrary dimension.

(2) $\mathcal{R}(P, O) = 0$ characterizes independence of P and O.

Distance correlation satisfies $0 \leq \mathcal{R} \leq 1$, and $\mathcal{R} = 0$ only if P and O are independent. Distance covariance $\mathcal{V}$ provides a new approach to the problem of testing the joint independence of random vectors. The formal definitions of the population coefficients $\mathcal{V}$ and $\mathcal{R}$ are given in Szekely et al. (2007).

The empirical distance correlation $\mathcal{R}_n(\mathbf{P}, \mathbf{O})$ is the square root of

$$\mathcal{R}_n^2(\mathbf{P}, \mathbf{O}) = \frac{\mathcal{V}_n^2(\mathbf{P}, \mathbf{O})}{\sqrt{\mathcal{V}_n^2(\mathbf{P})\mathcal{V}_n^2(\mathbf{O})}}.$$

For the formula and more details, see online-documentation and the energy-package

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Szekely, G.J., Rizzo, M.L., and Bakirov, N.K. (2007). *Measuring and testing dependence by correaltion of distances. Annals of Statistics, Vol. 35(6): 2769-2794*. doi:10.1214/009053607000000505.

Rizzo, M., and Szekely, G. (2022). energy: E-Statistics: Multivariate Inference via the Energy of Data. *R package version 1.7-10.* `https://CRAN.R-project.org/package=energy`.

See Also

eval_tidy, defusing-advanced dcor, energy

Examples

```
set.seed(1)
P <- rnorm(n = 100, mean = 0, sd = 10)
O <- P + rnorm(n=100, mean = 0, sd = 3)
dcorr(obs = P, pred = O)
```

---

deltap *deltaP or Markedness*

Description

deltap estimates the Markedness or deltaP for a nominal/categorical predicted-observed dataset.

mk estimates the Markedness (equivalent to deltaP) for a nominal/categorical predicted-observed dataset.

Usage

```
deltap(
  data = NULL,
  obs,
  pred,
  pos_level = 2,
  atom = FALSE,
  tidy = FALSE,
  na.rm = TRUE
)

mk(
  data = NULL,
  obs,
  pred,
  pos_level = 2,
  atom = FALSE,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).

pred	Vector with predicted values (character | factor).
pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.
atom	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE. When dataset is "binomial" atom does not apply.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The deltap is also known as the markedness. It is a metric that quantifies the probability that a condition is marked by the predictor with respect to a random chance (Powers, 2011).

The deltap is related to precision (or positive predictive values -ppv-) and its inverse (the negative predictive value -npv-) as follows:

$$deltap = PPV + NPV - 1 = precision + npv - 1$$

The higher the deltap the better the classification performance.

For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

References

Powers, D.M.W. (2011). Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies 2(1): 37–63.* doi:10.48550/arXiv.2010.16061

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))
# Get deltap estimate for two-class case
deltap(data = binomial_case, obs = labels, pred = predictions)
```

density_plot *Density plot of predicted and observed values*

Description

It draws a density area plot of predictions and observations with alternative axis orientation (P vs. O; O vs. P).

Usage

```
density_plot(
  data = NULL,
  obs,
  pred,
  n = 10,
  colors = c(low = NULL, high = NULL),
  orientation = "PO",
  print_metrics = FALSE,
  metrics_list = NULL,
  position_metrics = c(x = NULL, y = NULL),
  print_eq = TRUE,
  position_eq = c(x = NULL, y = NULL),
  eq_color = NULL,
  regline_type = NULL,
  regline_size = NULL,
  regline_color = NULL,
  na.rm = TRUE
)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
n	Argument of class numeric specifying the number of data points in each group.
colors	Vector or list with two colors '(low, high)' to paint the density gradient.
orientation	Argument of class string specifying the axis orientation, PO for predicted vs observed, and OP for observed vs predicted. Default is orientation = "PO".
print_metrics	boolean TRUE/FALSE to embed metrics in the plot. Default is FALSE.
metrics_list	vector or list of selected metrics to print on the plot.
position_metrics	vector or list with '(x,y)' coordinates to locate the metrics_table into the plot. Default : c(x = min(obs), y = 1.05*max(pred)).
print_eq	boolean TRUE/FALSE to embed metrics in the plot. Default is FALSE.

position_eq	vector or list with '(x,y)' coordinates to locate the SMA equation into the plot. Default : c(x = 0.70 max(x), y = 1.25*min(y)).
eq_color	string indicating the color of the SMA-regression text.
regline_type	string or integer indicating the SMA-regression line-type.
regline_size	number indicating the SMA-regression line size.
regline_color	string indicating the SMA-regression line color.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

It creates a density plot of predicted vs. observed values. The plot also includes the 1:1 line (solid line) and the linear regression line (dashed line). By default, it places the observed on the x-axis and the predicted on the y-axis (orientation = "PO"). This can be inverted by changing the argument orientation = "OP". For more details, see online-documentation

Value

Object of class ggplot.

See Also

ggplot,geom_point,aes

Examples

```
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 10)
density_plot(obs = X, pred = Y)
```

E1 *Absolute Model Efficiency (E1)*

Description

It estimates the E1 model efficiency using absolute differences.

Usage

```
E1(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The E1 is a type of model efficiency that modifies the Nash-Sutcliffe model efficiency by using absolute residuals instead of squared residuals. The E1 is used to overcome the NSE over-sensitivity to extreme values caused by the used of squared residuals. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Legates & McCabe (1999). Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. *Water Resour. Res.* doi:10.1029/1998WR900018

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
E1(obs = X, pred = Y)
```

`Erel` *Relative Model Efficiency (Erel)*

Description

It estimates the Erel model efficiency using differences to observations.

Usage

```
Erel(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The Erel model efficiency normalizes both residuals (numerator) and observed deviations (denominator) by observed values before squaring them. Compared to the NSE, the Erel is suggested as more sensitive to systematic over- or under-predictions. For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

References

Krause et al. (2005). Comparison of different efficiency criteria for hydrological model assessment. *Adv. Geosci. 5, 89–97.* doi:10.5194/adgeo5892005

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
Erel(obs = X, pred = Y)
```

---

error_rate *Error rate*

Description

It estimates the error rate for a nominal/categorical predicted-observed dataset.

Usage

```
error_rate(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list (default).
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The error rate represents the opposite of accuracy, referring to a measure of the degree to which the predictions miss-classify the reality. The classification error_rate is calculated as the ratio between the number of incorrectly classified objects with respect to the total number of objects. It is bounded between 0 and 1. The closer to 1 the worse. Values towards zero indicate low error_rate of predictions. It can be also expressed as percentage if multiplied by 100. It is estimated at a global level (not at the class level). The error rate is directly related to the accuracy, since error_rate = 1 – accuracy' . For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

(2017) Accuracy. *In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining Springer, Boston, MA.* doi:10.1007/9781489976871_3

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100,
replace = TRUE), predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100,
replace = TRUE), predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE)    )

# Get error_rate estimate for two-class case
error_rate(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get error_rate estimate for multi-class case
error_rate(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)
```

`fmi` *Fowlkes-Mallows Index*

Description

It estimates the Fowlkes-Mallows Index for a nominal/categorical predicted-observed dataset.

Usage

```
fmi(data = NULL, obs, pred, pos_level = 2, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (character | factor).
`pred`	Vector with predicted values (character | factor).
`pos_level`	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are `(Negative | Positive)`, `(0 | 1)`, `(FALSE | TRUE)`. Default : 2.
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The fmi has gained popularity within the machine learning community to summarize into a single value the confusion matrix of a binary classification. It is particularly useful when the number of observations belonging to each class is uneven or imbalanced. It is characterized for being symmetric (i.e. no class has more relevance than the other). It is bounded between -1 and 1. The closer to 1 the better the classification performance.

The fmi is only available for the evaluation of binary cases (two classes). For multiclass cases, fmi will produce a `NA` and display a warning.

For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Fowlkes, Edward B; Mallows, Colin L (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association. 78 (383): 553–569.* doi:10.1080/01621459.1983.10478008

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))
# Get fmi estimate for two-class case
fmi(data = binomial_case, obs = labels, pred = predictions)
```

`fscore` *F-score*

Description

It estimates the F-score for a nominal/categorical predicted-observed dataset.

Usage

```
fscore(
  data = NULL,
  obs,
  pred,
  B = 1,
  pos_level = 2,
  atom = FALSE,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (character | factor).
`pred`	Vector with predicted values (character | factor).
`B`	Numeric value indicating the weight (a.k.a. B or beta) to be applied to the relationship between recall and precision. B < 1 weights more precision than recall. B > 1 gives B times more importance to recall than precision. Default: 1.
`pos_level`	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are `(Negative | Positive)`, `(0 | 1)`, `(FALSE | TRUE)`. Default : 2.
`atom`	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE. When dataset is "binomial" atom does not apply.
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.

na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The F-score (or F-measure) it is a more robust metric than the classic accuracy, especially when the number of cases for each class is uneven. It represents the harmonic mean of precision and recall. Thus, to achieve high values of F-score it is necessary to have both high precision and high recall.

The universal version of F-score employs a coefficient B, by which we can tune the precision-recall ratio. Values of B > 1 give more weight to recall, and B < 1 give more weight to precision.

For binomial/binary cases, fscore = TP / (TP + 0.5*(FP + FN))

The generalized formula applied to multiclass cases is:

$fscore = \frac{(1+B^2)*(precision*recall)}{((B^2*precision)+recall)}$

It is bounded between 0 and 1. The closer to 1 the better. Values towards zero indicate low performance. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Goutte, C., Gaussier, E. (2005). A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. *In: D.E. Losada and J.M. Fernandez-Luna (Eds.): ECIR 2005 . Advances in Information Retrieval LNCS 3408, pp. 345–359, 2. Springer-Verlag Berlin Heidelberg.* doi:10.1007/9783540318651_25

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100, replace = TRUE),
predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE)    )

# Get F-score estimate for two-class case
fscore(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get F-score estimate for each class for the multi-class case
fscore(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get F-score estimate for the multi-class case at a global level
fscore(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)
```

gmean *Geometric Mean*

Description

It estimates the Geometric Mean score for a nominal/categorical predicted-observed dataset.

Usage

```
gmean(
  data = NULL,
  obs,
  pred,
  pos_level = 2,
  atom = FALSE,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.
atom	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE. When dataset is "binomial" atom does not apply.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The gmean is a metric especially useful for imbalanced classes because it measures the balance between the classification performance on both major (over-represented) as well as on minor (under-represented) classes. As stated above, it is particularly useful when the number of observations belonging to each class is uneven.

The gmean score is equivalent to the square-root of the product of specificity and recall (a.k.a. sensitivity).

$gmean = \sqrt{recall * specificity}$

It is bounded between 0 and 1. The closer to 1 the better the classification performance, while zero represents the worst.

For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

De Diego, I.M., Redondo, A.R., Fernández, R.R., Navarro, J., Moguerza, J.M. (2022). General Performance Score for classification problems. _ Appl. Intell. (2022)._ doi:10.1007/s10489021-030417

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))
# Get gmean estimate for two-class case
gmean(data = binomial_case, obs = labels, pred = predictions)
```

`import_apsim_db` *Import SQLite databases generated by APSIM NextGen*

Description

Imports data from SQLite databases (*.db) applied by APSIM Next Generation. It reads one file at a time.

Usage

```
import_apsim_db(filename = "", folder = ".", value = "report", simplify = TRUE)
```

Arguments

`filename`	file name including the file extension ("*.db"), as a string ("").
`folder`	source folder/directory containing the file, as a string ("").
`value`	either 'report', 'all' (list) or user-defined for a specific report.
`simplify`	if TRUE will attempt to simplify multiple reports into a single data.frame. If FALSE it will return a list. Default: TRUE.

Value

An object of class `data.frame`, but it depends on the argument 'value' above

Note

This function was adapted from apsimx package (Miguez, 2022). For reference, we recommend to check and use the function apsimx::read_apsimx() as an alternative. Source: https://github.com/femiguez/apsimx by F. Miguez.

References

Miguez, F. (2022) apsimx: Inspect, Read, Edit and Run 'APSIM' "Next Generation" and 'APSIM' Classic. *R package version 2.3.1,* https://CRAN.R-project.org/package=apsimx

Examples

```
## See [documentation](https://adriancorrendo.github.io/metrica/index.html)
```

`import_apsim_out` *import_apsim_out*

Description

Function to import *.out files generated by APSIM Classic.

Usage

```
import_apsim_out(filepath)
```

Arguments

`filepath`	Argument to specify the location path to the *.out file.

Details

This function will generate a data frame object from an APSIM Classic *.out file.

Value

Element of class `data.frame`.

Note

Note: It imports one file at a time.

Examples

```
## See [documentation](https://adriancorrendo.github.io/metrica/index.html)
```

iqRMSE *Inter-Quartile Root Mean Squared Error*

Description

It estimates the IqRMSE for a continuous predicted-observed dataset.

Usage

```
iqRMSE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The iqRMSE normalizes the RMSE by the length of the inter-quartile range of observations (percentiles 25th to 75th). As an error metric, the lower the values the better. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
iqRMSE(obs = X, pred = Y)
```

KGE *Kling-Gupta Model Efficiency (KGE).*

Description

It estimates the KGE for a predicted-observed dataset.

Usage

```
KGE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The KGE is a normalized, dimensionless, model efficiency that measures general agreement. It presents accuracy, precision, and consistency components. It is symmetric (invariant to predicted observed orientation). It is positively bounded up to 1. The closer to 1 the better. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Kling et al. (2012). Runoff conditions in the upper Danube basin under an ensemble of climate change scenarios. *Journal of Hydrology 424-425, 264-277.* doi:10.1016/j.jhydrol.2012.01.011

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
KGE(obs = X, pred = Y)
```

khat *K-hat (Cohen's Kappa Coefficient)*

Description

It estimates the Cohen's Kappa Coefficient for a nominal/categorical predicted-observed dataset.

Usage

```
khat(data = NULL, obs, pred, pos_level = 2, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The Cohen's Kappa Coefficient is the accuracy normalized by the possibility of agreement by chance. Thus, it is considered a more robust agreement measure than simply the accuracy. The kappa coefficient was originally described for evaluating agreement of classification between different "raters" (inter-rater reliability).

It is positively bounded to 1, but it is not negatively bounded. The closer to 1 the better as Kappa assumes its theoretical maximum value of 1 (perfect agreement) only when both observed and predicted values are equally distributed across the classes (i.e. identical row and column sums). Thus, the lower the kappa the lower the prediction quality.

For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

References

Cohen, J. (1960). A coefficient of agreement for nominal scales. _ Educational and Psychological Measurement 20 (1): 37–46._ doi:10.1177/001316446002000104

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100,
replace = TRUE), predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE))

# Get Cohen's Kappa Coefficient estimate for two-class case
khat(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get Cohen's Kappa Coefficient estimate for each class for the multi-class case
khat(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get Cohen's Kappa Coefficient estimate for the multi-class case at a global level
khat(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)
```

---

lambda *Duveiller's Agreement Coefficient*

Description

It estimates the agreement coefficient (lambda) suggested by Duveiller et al. (2016) for a continuous predicted-observed dataset.

Usage

```
lambda(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

lambda measures both accuracy and precision. It is normalized, dimensionless, bounded (-1;1), and symmetric (invariant to predicted-observed orientation). lambda is equivalent to CCC when r is greater or equal to 0. The closer to 1 the better. Values towards zero indicate low correlation between observations and predictions. Negative values would indicate a negative relationship between predicted and observed. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Duveiller et al. (2016). Revisiting the concept of a symmetric index of agreement for continuous datasets. *Sci. Rep. 6, 1-14.* doi:10.1038/srep19401

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
lambda(obs = X, pred = Y)
```

---

land_cover *Binary Land Cover Data*

Description

This example dataset was obtained in 2022 over a small region in Kansas, using visual interpretation. The predicted values were the result of a Random Forest classification, with a 70/30 % split. Values equal to 1 are associated to vegetation, and values equal to 0 are other type of land cover.

Usage

```
land_cover
```

Format

This data frame has 284 rows and the following 2 columns:

predicted predicted values

actual observed values

Source

https://github.com/adriancorrendo/metrica

LCS *Lack of Correlation (LCS)*

Description

It estimates the lack of correlation (LCS) component of the Mean Squared Error (MSE) proposed by Kobayashi & Salam (2000).

Usage

```
LCS(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The LCS represents the random component of the prediction error following Kobayashi & Salam (2000). The lower the value the less contribution to the MSE. However, it needs to be compared to MSE as its benchmark. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Kobayashi & Salam (2000). Comparing simulated and measured values using mean squared deviation and its components. *Agron. J. 92, 345–352.* doi:10.2134/agronj2000.922345x

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
LCS(obs = X, pred = Y)
```

`likelihood_ratios` *Likelihood Ratios (Classification)*

Description

It estimates the positive likelihood ratio `posLr`, negative likelihood ratio `negLr`, and diagnostic odds ratio `dor` for a nominal/categorical predicted-observed dataset.

Usage

```
posLr(
  data = NULL,
  obs,
  pred,
  pos_level = 2,
  atom = FALSE,
  tidy = FALSE,
  na.rm = TRUE
)

negLr(
  data = NULL,
  obs,
  pred,
  pos_level = 2,
  atom = FALSE,
  tidy = FALSE,
  na.rm = TRUE
)

dor(
  data = NULL,
  obs,
  pred,
  pos_level = 2,
  atom = FALSE,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (character | factor).
`pred`	Vector with predicted values (character | factor).

pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.
atom	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE. When dataset is "binomial" atom does not apply.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The likelihood ratios are metrics designed to assess the expectations of classification tasks. They are highly related to recall (sensitivity or true positive rate) and specificity (selectivity or true negative rate).

For a multiclass case, positive and negative results are class-wise. We can either hit the actual class, or the actual non-class. For example, a classification may have 3 potential results: cat, dog, or fish. Thus, when the actual value is dog, the only true positive is dog, but we can obtain a true negative either by classifying it as a cat or a fish. The posLr, negLr, and dor estimates are the mean across all classes.

The positive likelihood ratio (posLr) measures the odds of obtaining a positive prediction in cases that are actual positives.

The negative likelihood ratio (negLr) relates the odds of obtaining a negative prediction for actual non-negatives relative to the probability of actual negative cases obtaining a negative prediction result.

Finally, the diagnostic odds ratio (dor) measures the effectiveness of classification. It represents the odds of a positive prediction result in actual (observed) positive cases with respect to the odds of a positive prediction for the actual negative cases.

The ratios are define as follows:

$posLr = \frac{recall}{1+specificity} = \frac{TPR}{FPR}$

$negLr = \frac{1-recall}{specificity} = \frac{FNR}{TNR}$

$dor = \frac{posLr}{negLr}$

For more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

References

GlasaJeroen, A.S., Lijmer, G., Prins, M.H., Bonsel, G.J., Bossuyta, P.M.M. (2009). The diagnostic odds ratio: a single indicator of test performance. *Journal of Clinical Epidemiology 56(11): 1129-1135.* doi:10.1016/S08954356(03)00177X

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100, replace = TRUE),
predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE)    )

# Get posLr, negLr, and dor for a two-class case
posLr(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)
negLr(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)
dor(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)
```

MAE *Mean Absolute Error (MAE)*

Description

It estimates the MAE for a continuous predicted-observed dataset.

Usage

```
MAE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The MAE measures both lack of accuracy and precision in absolute scale. It keeps the same units than the response variable. It is less sensitive to outliers than the MSE or RMSE. The lower the better. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Willmott & Matsuura (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim. Res. 30, 79–82.* doi:10.3354/cr030079

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
MAE(obs = X, pred = Y)
```

`maize_phenology` *Multi Class Phenology*

Description

This example data set includes maize phenology classes collected in Kansas during the 2018 growing season. The predicted values were obtained using a Random Forest classifier with a 70/30 split. The dataset includes 16 phenology stages from VT to R6. For more information please visit doi:10.3390/rs14030469.

Usage

```
maize_phenology
```

Format

This data frame has 103 rows and the following 2 columns:

predicted predicted values

actual observed values

Source

https://github.com/adriancorrendo/metrica

MAPE *Mean Absolute Percentage Error (MAPE)*

Description

It estimates the MAPE of a continuous predicted-observed dataset.

Usage

```
MAPE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The MAPE is expressed in percentage units (independent scale), which it makes easy to explain and to compare performance across models with different response variables. MAPE is asymmetric (sensitive to axis orientation). The lower the better. As main disadvantage, MAPE produces infinite or undefined values for zero or close-to-zero observed values. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Kim & Kim (2016). A new metric of absolute percentage error for intermittent demand forecast. _Int. J. Forecast. 32, 669-679._doi:10.1016/j.ijforecast.2015.12.003

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
MAPE(obs = X, pred = Y)
```

MASE *Mean Absolute Scaled Error (MASE)*

Description

It estimates the mean absolute error using the naive-error approach for a continuous predicted-observed dataset.

Usage

```
MASE(
  data = NULL,
  obs,
  pred,
  time = NULL,
  naive_step = 1,
  oob_mae = NULL,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

data	(Required) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
time	(Optional) String with the "name" of the vector containing the time variable to sort observations. "Optional" to ensure an appropriate MASE estimation. Default: NULL, assumes observations are already sorted by time.
naive_step	A positive number specifying how many observed values to recall back in time when computing the naive expectation. Default = 1
oob_mae	A numeric value indicating the out-of-bag (out-of-sample) MAE. By default, an *in-sample* MAE is calculated using the naive forecast method. See Hyndman & Koehler (2006). Default : NULL.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The MASE is especially well suited for time series predictions. It can be used to compare forecast methods on a single series and also to compare forecast accuracy between series.

This metric never gives an infinite or undefined values (unless all observations over time are exactly the same!).

By default, the MASE scales the error based on *in-sample* MAE from the naive forecast method (random walk). The in-sample MAE is used in the denominator because it is always available and it effectively scales the errors.Since it is based on absolute error, it is less sensitive to outliers compared to the classic MSE.

$MASE = \frac{1}{n}(\frac{|O_i - P_i|}{\frac{1}{T-1}\sum_{t=2}^{T}|O_t - O_{t-1}|})$

If available, users may use and out-of-bag error from an independent dataset, which can be specified with the `oob_mae` arg. and will replace the denominator into the MASE equation.

MASE measures total error (i.e. both lack of accuracy and precision.). The lower the MASE below 1, the better the prediction quality. MASE = indicates no difference between the model and the naive forecast (or oob predictions). MASE > 1 indicate poor performance.

For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Hyndman, R.J., Koehler, A.B. (2006). Another look at measures of forecast accuracy. _ Int. J. Forecast_ doi:10.3354/cr030079

Examples

```
set.seed(1)
# Create a fake dataset
X <- rnorm(n = 100, mean = 8, sd = 10)
Y <- rnorm(n = 100, mean = 8.2, sd = 15)
Z <- seq(1, 100, by = 1)

time_data <- as.data.frame(list("observed" = X, "predicted" = Y, "time" = Z))

MASE(data = time_data, obs = observed, pred = predicted, time = time)
```

MBE *Mean Bias Error (MBE)*

Description

It estimates the MBE for a continuous predicted-observed dataset.

Usage

```
MBE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The MBE is one of the most widely used error metrics. It presents the same units than the response variable, and it is unbounded. It can be simply estimated as the difference between the means of predictions and observations. The closer to zero the better. Negative values indicate overestimation. Positive values indicate general underestimation. The disadvantages are that is only sensitive to additional bias, so the MBE may mask a poor performance if overestimation and underestimation co-exist (a type of proportional bias). For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
MBE(obs = X, pred = Y)
```

---

mcc *Matthews Correlation Coefficient* | *Phi Coefficient*

Description

It estimates the `mcc` for a nominal/categorical predicted-observed dataset.

`phi_coef` estimates the Phi coefficient (equivalent to the Matthews Correlation Coefficient `mcc`).

Usage

```
mcc(data = NULL, obs, pred, pos_level = 2, tidy = FALSE, na.rm = TRUE)

phi_coef(data = NULL, obs, pred, pos_level = 2, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The mcc it is also known as the phi-coefficient. It has gained popularity within the machine learning community to summarize into a single value the confusion matrix of a binary classification.

It is particularly useful when the number of observations belonging to each class is uneven or imbalanced. It is characterized for being symmetric (i.e. no class has more relevance than the other). It is bounded between -1 and 1. The closer to 1 the better the classification performance.

For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

References

Chicco, D., Jurman, G. (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics 21, 6 (2020).* doi:10.1186/s1286401964137

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))

# Get mcc estimate for two-class case
mcc(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)
```

`metrics_summary` *Prediction Performance Summary*

Description

It estimates a group of metrics characterizing the prediction performance for a continuous (regression) or categorical (classification) predicted-observed dataset. By default, it calculates all available metrics for either regression or classification.

Usage

```
metrics_summary(
  data = NULL,
  obs,
  pred,
  type = NULL,
  metrics_list = NULL,
  orientation = "PO",
  pos_level = 2,
  na.rm = TRUE
)
```

Arguments

`data`	argument to call an existing data frame containing the data (optional).
`obs`	vector with observed values (numeric).
`pred`	vector with predicted values (numeric).
`type`	argument of class string specifying the type of model. For continuous variables use *type = 'regression'*. For categorical variables use *type = 'classification'*.
`metrics_list`	vector or list of specific selected metrics. Default is = NULL, which will estimate all metrics available for either regression or classification.
`orientation`	argument of class string specifying the axis orientation to estimate slope(B1) and intercept(B0). It only applies when type = "regression". "PO" is for predicted vs observed, and "OP" for observed vs predicted. Default is orientation = "PO".
`pos_level`	(for classification only). Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are `(Negative | Positive)`, `(0 | 1)`, `(FALSE | TRUE)`. Default : 2.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The user can choose to calculate a single metric, or to calculate all metrics at once. This function creates a data.frame with all (or selected) metrics in the `metrica`-package. If looking for specific metrics, the user can pass a list of desired metrics using the argument "metrics_list" (e.g. metrics_list = c("R2","MAE", "RMSE", "RSR", "NSE", "KGE")). For the entire list of available metrics with formula, see online-documentation

Value

an object of class `data.frame` containing all (or selected) metrics.

Examples

```
# Continuous variable (regression)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 10)
regression_case <- data.frame(obs = X, pred = Y)

# Get a metrics summary for a regression problem
metrics_summary(regression_case, obs = X, pred = Y, type = "regression")

# Categorical variable (classification)
binomial_case <- data.frame(labels = sample(c("True","False"), 100,
replace = TRUE), predictions = sample(c("True","False"), 100, replace = TRUE))

#' # Get a metrics summary for a regression problem
metrics_summary(binomial_case, obs = labels, pred = predictions,
type = "classification")
```

MIC *Maximal Information Coefficient*

Description

It estimates the Maximal Information Coefficient (MIC) for a continuous predicted-observed dataset.

Usage

```
MIC(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list (default).

`na.rm` Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The **MIC** function is a wrapper for the `mine_stat` function of the **minerva**-package, a collection of Maximal Information-Based Nonparametric statistics (MINE). See Reshef et al. (2011).

For the predicted-observed case (PO), the **MIC** is defined as follows:

$$\text{MIC}(D) = \max_{PO<B(n)} M(D)_{X,Y} = \max_{PO<B(n)} \frac{I^*(D,P,O)}{log(\min P,O)},$$

where $B(n) = n^\alpha$ is the search-grid size, $I^*(D,P,O)$ is the maximum mutual information over all grids *P-by-O*, of the distribution induced by D on a grid having *P* and *O* bins (where the probability mass on a cell of the grid is the fraction of points of D falling in that cell). Albanese et al. (2013).

For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Reshef, D., Reshef, Y., Finucane, H., Grossman, S., McVean, G., Turnbaugh, P., Lander, R., Mitzenmacher, M., and Sabeti, P. (2011). Detecting novel associations in large datasets. *Science 334, 6062*. doi:10.1126/science.1205438.

Albanese, D., M. Filosi, R. Visintainer, S. Riccadonna, G. Jurman, C. Furlanello. minerva and minepy: a C engine for the MINE suite and its R, Python and MATLAB wrappers. *Bioinformatics (2013) 29(3):407-408*. doi:10.1093/bioinformatics/bts707.

See Also

`eval_tidy`, `defusing-advanced` `mine_stat`

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
MIC(obs = X, pred = Y)
```

MLA *Mean Lack of Accuracy (MLA)*

Description

It estimates the MLA, the systematic error component to the Mean Squared Error (MSE), for a continuous predicted-observed dataset following Correndo et al. (2021).

Usage

```
MLA(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The MLA represents the systematic (bias) component of the MSE. It is obtained via a symmetric decomposition of the MSE (invariant to predicted-observed orientation) using a symmetric regression line. The MLA is equal to the sum of systematic differences divided by the sample size (n). The greater the value the greater the bias of the predictions. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Correndo et al. (2021). Revisiting linear regression to test agreement in continuous predicted-observed datasets. *Agric. Syst. 192, 103194.* doi:10.1016/j.agsy.2021.103194

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
MLA(obs = X, pred = Y)
```

MLP *Mean Lack of Precision (MLP)*

Description

It estimates the MLP, the unsystematic error component to the Mean Squared Error (MSE), for a continuous predicted-observed dataset following Correndo et al. (2021).

Usage

```
MLP(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The MLP represents the unsystematic (random) component of the MSE. It is obtained via a symmetric decomposition of the MSE (invariant to predicted-observed orientation) using a symmetric regression line. The MLP is equal to the sum of unsystematic differences divided by the sample size (n). The greater the value the greater the random noise of the predictions. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Correndo et al. (2021). Revisiting linear regression to test agreement in continuous predicted-observed datasets. *Agric. Syst. 192, 103194.* doi:10.1016/j.agsy.2021.103194

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
MLP(obs = X, pred = Y)
```

MSE *Mean Squared Error (MSE)*

Description

It estimates the MSE for a continuous predicted-observed dataset.

Usage

```
MSE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The MSE, also known as MSD, measures general agreement, as includes both variance (lack of precision) and bias (lack of accuracy). The MSE of predictions could be decomposed following a variety of approaches (e.g. Willmott et al. 1981; Correndo et al. 2021). Its calculation is simple, the sum of squared differences between predictions and observations divided by the sample size (n). The greater the value the worse the predicted performance. Unfortunately, the units of MSE do not have a direct interpretation. For a more direct interpretation, the square root of MSE (RMSE) has the same units as the variable of interest. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Willmott (1981). On the validation of models. *Phys. Geogr. 2, 184–194.* doi:10.1080/02723646.1981.10642213

Correndo et al. (2021). Revisiting linear regression to test agreement in continuous predicted-observed datasets. *Agric. Syst. 192, 103194.* doi:10.1016/j.agsy.2021.103194

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
MSE(obs = X, pred = Y)
```

---

npv *Negative Predictive Value*

Description

npv estimates the npv (a.k.a. positive predictive value -PPV-) for a nominal/categorical predicted-observed dataset.

FOR estimates the false omission rate, which is the complement of the negative predictive value for a nominal/categorical predicted-observed dataset.

Usage

```
npv(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)

FOR(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
atom	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE.

pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The npv is a non-normalized coefficient that represents the ratio between the correctly predicted cases (or true positive -TP- for binary cases) to the total predicted observations for a given class (or total predicted positive -PP- for binary cases) or at overall level.

For binomial cases, $npv = \frac{TP}{PP} = \frac{TP}{TP+FP}$

The npv metric is bounded between 0 and 1. The closer to 1 the better. Values towards zero indicate low npv of predictions. It can be estimated for each particular class or at a global level.

The false omission rate (FOR) represents the proportion of false negatives with respect to the number of negative predictions (PN).

For binomial cases, $FOR = 1 - npv = \frac{FN}{PN} = \frac{FN}{TN+FN}$

The npv metric is bounded between 0 and 1. The closer to 1 the better. Values towards zero indicate low npv of predictions.

For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

References

Wang H., Zheng H. (2013). Negative Predictive Value. *In: Dubitzky W., Wolkenhauer O., Cho KH., Yokota H. (eds) Encyclopedia of Systems Biology.* _ Springer, New York, NY._ doi:10.1007/9781441998637_234

Trevethan, R. (2017). *Sensitivity, Specificity, and Predictive Values: Foundations, Pliabilities, and Pitfalls* _ in Research and Practice. Front. Public Health 5:307_ doi:10.3389/fpubh.2017.00307

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100,
replace = TRUE), predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100,
replace = TRUE), predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE))
```

```
# Get npv estimate for two-class case
npv(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get fdr estimate for two-class case
FDR(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get npv estimate for each class for the multi-class case
npv(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE, atom = TRUE)

# Get npv estimate for the multi-class case at a global level
npv(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE, atom = TRUE)
```

NSE *Nash-Sutcliffe Model Efficiency (NSE)*

Description

It estimates the model efficiency suggested by Nash & Sutcliffe (1970) for a continuous predicted-observed dataset.

Usage

```
NSE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The NSE measures general agreement. It is normalized (by the variance of the observations) and dimensionless. It is calculated using the absolute squared differences between the predictions and observations, which has been suggested as an issue due to over-sensitivity to outliers. It goes form -infinity to 1. The closer to 1 the better the prediction performance. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Nash & Sutcliffe (1970). River flow forecasting through conceptual models part I - A discussion of principles. *J. Hydrol. 10(3), 292-290.* doi:10.1016/00221694(70)902556

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
NSE(obs = X, pred = Y)
```

---

p4 *P4-metric*

Description

It estimates the P4-metric for a nominal/categorical predicted-observed dataset.

Usage

```
p4(
  data = NULL,
  obs,
  pred,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE,
  atom = FALSE
)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.
atom	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE. When dataset is "binomial" atom does not apply.

Details

The P4-metric it is a metric designed for binary classifiers. It is estimated from precision, recall, specificity, and npv (negative predictive value). The P4 it was designed to address criticism against the F-score, so it may be perceived as its extension. Unfortunately, it has not been generalized yet for multinomial cases.

For binomial/binary cases,

$p4 = \frac{(4xTPxTN)}{(4xTPxTN+(TP+TN)xFP+FN)}$

Or:

$p4 = \frac{4}{\frac{1}{precision}+\frac{1}{recall}+\frac{1}{specificity}+\frac{1}{npv}}$

The P4 metric has not been generalized for multinomial cases. The P4 metric is bounded between 0 and 1. The closer to 1 the better, which will require precision, recall, specificity and npv being close to 1. Values towards zero indicate low performance, which could be the product of only one of the four conditional probabilities being close to 0.

For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Sitarz, M. (2023). Extending F1 metric, probabilistic approach. _Adv. Artif. Intell. Mach. Learn., 3 (2):1025-1038._doi:10.54364/AAIML.2023.1161

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100, replace = TRUE),
predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE)    )

# Get P4-metric estimate for two-class case
p4(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)
```

PAB *Percentage Additive Bias (PAB)*

Description

It estimates the contribution of the proportional bias to the Mean Squared Error (MSE) following Correndo et al. (2021).

Usage

```
PAB(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The PAB represents contribution of additive bias () component of the MSE. The PAB = 100*((mO-mP)^2/MSE), where mO, and mP represent the mean of observations and predictions, respectively. The greater the value the greater the contribution of additive bias to the prediction error. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Correndo et al. (2021). Revisiting linear regression to test agreement in continuous predicted-observed datasets. *Agric. Syst. 192, 103194.* doi:10.1016/j.agsy.2021.103194

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
PAB(obs = X, pred = Y)
```

PBE *Percentage Bias Error (PBE).*

Description

It estimates the PBE for a continuous predicted-observed dataset following Gupta et al. (1999).

Usage

```
PBE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The PBE (%) is useful to identify systematic over or under predictions. It is an unbounded metric. The closer to zero the bias of predictions. Negative values indicate overestimation, while positive values indicate underestimation. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Gupta et al. (1999). Status of automatic calibration for hydrologic models: Comparison with multi-level expert calibration. *J. Hydrologic Eng. 4(2): 135-143.* doi:10.1061/(ASCE)10840699(1999)4:2(135)

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
PBE(obs = X, pred = Y)
```

PLA *Percentage Lack of Accuracy (PLA)*

Description

It estimates the PLA, the contribution of the systematic error to the Mean Squared Error (MSE) for a continuous predicted-observed dataset following Correndo et al. (2021).

Usage

```
PLA(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The PLA (%, 0-100) represents the contribution of the Mean Lack of Accuracy (MLA), the systematic (bias) component of the MSE. It is obtained via a symmetric decomposition of the MSE (invariant to predicted-observed orientation). The PLA can be further segregated into percentage additive bias (PAB) and percentage proportional bias (PPB). The greater the value the greater the contribution of systematic error to the MSE. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Correndo et al. (2021). Revisiting linear regression to test agreement in continuous predicted-observed datasets. *Agric. Syst. 192, 103194.* doi:10.1016/j.agsy.2021.103194

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
PLA(obs = X, pred = Y)
```

PLP *Percentage Lack of Precision (PLP)*

Description

It estimates the PLP, the contribution of the unsystematic error to the Mean Squared Error (MSE) for a continuous predicted-observed dataset following Correndo et al. (2021).

Usage

```
PLP(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The PLP (%, 0-100) represents the contribution of the Mean Lack of Precision (MLP), the unsystematic (random) component of the MSE. It is obtained via a symmetric decomposition of the MSE (invariant to predicted-observed orientation). The greater the value the greater the contribution of unsystematic error to the MSE. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Correndo et al. (2021). Revisiting linear regression to test agreement in continuous predicted-observed datasets. *Agric. Syst. 192, 103194.* doi:10.1016/j.agsy.2021.103194

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
PLP(obs = X, pred = Y)
```

PPB *Percentage Proportional Bias (PPB)*

Description

It estimates the contribution of the proportional bias to the Mean Squared Error (MSE) following Correndo et al. (2021).

Usage

```
PPB(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The PPB (%) measures the contribution of proportional bias to the MSE. The PPB = 100*(((sO - sP)^2)/MSE), where sO, and sP are the sample variances of observations and predictions, respectively. The greater the value the greater the contribution of proportional bias to the prediction error. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Correndo et al. (2021). Revisiting linear regression to test agreement in continuous predicted-observed datasets. *Agric. Syst. 192, 103194.* doi:10.1016/j.agsy.2021.103194

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
PPB(obs = X, pred = Y)
```

precision *Precision* | *Positive Predictive Value*

Description

precision estimates the precision (a.k.a. positive predictive value -ppv-) for a nominal/categorical predicted-observed dataset.

ppv estimates the Positive Predictive Value (equivalent to precision) for a nominal/categorical predicted-observed dataset.

FDR estimates the complement of precision (a.k.a. positive predictive value -PPV-) for a nominal/categorical predicted-observed dataset.

Usage

```
precision(
  data = NULL,
  obs,
  pred,
  tidy = FALSE,
  atom = FALSE,
  na.rm = TRUE,
  pos_level = 2
)

ppv(
  data = NULL,
  obs,
  pred,
  tidy = FALSE,
  atom = FALSE,
  na.rm = TRUE,
  pos_level = 2
)

FDR(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.

obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
atom	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.
pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.

Details

The precision is a non-normalized coefficient that represents the ratio between the correctly predicted cases (or true positive -TP- for binary cases) to the total predicted observations for a given class (or total predicted positive -PP- for binary cases) or at overall level.

For binomial cases, $precision = \frac{TP}{PP} = \frac{TP}{TP+FP}$

The precision metric is bounded between 0 and 1. The closer to 1 the better. Values towards zero indicate low precision of predictions. It can be estimated for each particular class or at a global level.

The false detection rate or false discovery rate (FDR) represents the proportion of false positives with respect to the total number of cases predicted as positive.

For binomial cases, $FDR = 1 - precision = \frac{FP}{PP} = \frac{FP}{TP+FP}$

The precision metric is bounded between 0 and 1. The closer to 1 the better. Values towards zero indicate low precision of predictions.

For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

References

Ting K.M. (2017) Precision and Recall. *In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA.* doi:10.1007/97814899768 71_659

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100,
replace = TRUE), predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
```

```
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100,
replace = TRUE), predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE))

# Get precision estimate for two-class case
precision(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get FDR estimate for two-class case
FDR(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get precision estimate for each class for the multi-class case
precision(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE, atom = TRUE)

# Get precision estimate for the multi-class case at a global level
precision(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE, atom = TRUE)
```

`prevalence` *Prevalence*

Description

`preval` estimates the prevalence of positive cases for a nominal/categorical predicted-observed dataset.

`preval_t` estimates the prevalence threshold for a binary predicted-observed dataset.

Usage

```
preval(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)

preval_t(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
atom	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE.
pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The prevalence measures the overall proportion of actual positives with respect to the total number of observations. Currently, it is defined for binary cases only.

The general formula is:

$preval = \frac{positive}{positive+negative}$

The prevalence threshold represents an point on the ROC curve (function of sensitivity (recall) and specificity) below which the precision (or PPV) dramatically drops.

$preval_t = \frac{\sqrt{TPR*FPR}-FPR}{TPR-FPR}$

It is bounded between 0 and 1. The closer to 1 the better. Values towards zero indicate low performance. For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

References

Freeman, E.A., Moisen, G.G. (2008). A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa. . *Ecol. Modell. 217(1-2): 45-58.* doi:10.1016/j.ecolmodel.2008.05.015

Balayla, J. (2020). Prevalence threshold and the geometry of screening curves. _Plos one, 15(10):e0240215, _ doi:10.1371/journal.pone.0240215

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100, replace = TRUE),
```

```
predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100, replace = TRUE),
predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE)    )

# Get prevalence estimate for two-class case
preval(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get prevalence estimate for each class for the multi-class case
preval(data = multinomial_case, obs = labels, pred = predictions, atom = TRUE)
```

`r` *Sample Correlation Coefficient (r)*

Description

It estimates the Pearson's coefficient of correlation (r) for a continuous predicted-observed dataset.

Usage

```
r(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The r coefficient measures the strength of linear relationship between two variables. It only accounts for precision, but it is not sensitive to lack of prediction accuracy. It is a normalized, dimensionless coefficient, that ranges between -1 to 1. It is expected that predicted and observed values will show $0 < r < 1$. It is also known as the Pearson Product Moment Correlation, among other names. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Kirch (2008) Pearson's Correlation Coefficient. *In: Kirch W. (eds) Encyclopedia of Public Health. Springer, Dordrecht.* doi:10.1007/9781402056147_2569

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
r(obs = X, pred = Y)
```

R2 *Coefficient of determination (R2).*

Description

It estimates the R2 for a continuous predicted-observed dataset.

Usage

```
R2(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The R2 is one of the most widely used metrics evaluating models performance. R2 is appropriate to estimate the strength of linear association between two variables. It is positively bounded to 1, but it may produce negative values. The closer to 1 the better linear association between predictions and observations. However, R2 presents a major flaw for prediction performance evaluation: it is not sensitive to lack of accuracy (additive or proportional bias). Thus, R2 only measures precision, but it does not account for accuracy of the predictions. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Yang et al. (2014). An evaluation of the statistical methods for testing the performance of crop models with observed data. *Agric. Syst. 127, 81-89.* doi:10.1016/j.agsy.2014.01.008

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
R2(obs = X, pred = Y)
```

RAC *Robinson's Agreement Coefficient (RAC).*

Description

It estimates the agreement coefficient suggested by Robinson (1957; 1959) for a continuous predicted-observed dataset.

Usage

```
RAC(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The RAC measures both accuracy and precision (general agreement). It is normalized, dimensionless, bounded (0 to 1), and symmetric (invariant to predicted-observed orientation). For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Robinson (1957). The statistical measurement of agreement. *Am. Sociol. Rev. 22(1), 17-25* doi:10.2307/2088760

Robinson (1959). The geometric interpretation of agreement. *Am. Sociol. Rev. 24(3), 338-345* doi:10.2307/2089382

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
RAC(obs = X, pred = Y)
```

RAE *Relative Absolute Error (RAE)*

Description

It estimates the RAC for a continuous predicted-observed dataset.

Usage

```
RAE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The RAE normalizes the Mean Absolute Error (MAE) with respect to the total absolute error. It is calculated as the ratio between the sum of absolute residuals (error of predictions with respect to observations) and the sum of absolute errors of observations with respect to its mean. It presents its lower bound at 0 (perfect fit), and has no upper bound. It can be used to compare models using different units. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
RAE(obs = X, pred = Y)
```

---

recall *Recall* | *Sensitivity* | *True Positive Rate* | *Hit rate*

Description

recall estimates the recall (a.k.a. sensitivity, true positive rate -TPR-, or hit rate) for a nominal/categorical predicted-observed dataset.

TPR alternative to recall().

sensitivity alternative to recall().

hitrate alternative to recall().

FNR estimates false negative rate (or false alarm, or fall-out) for a nominal/categorical predicted-observed dataset.

Usage

```
recall(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)

TPR(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)

sensitivity(
  data = NULL,
  obs,
```

```
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)

hitrate(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)

FNR(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
atom	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE. When dataset is "binomial" atom does not apply.
pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The recall (a.k.a. sensitivity or true positive rate -TPR-) is a non-normalized coefficient that represents the ratio between the correctly predicted cases (true positives -TP-) to the total number

of actual observations that belong to a given class (actual positives -P-).

For binomial cases, $recall = \frac{TP}{P} = \frac{TP}{TP+FN}$

The `recall` metric is bounded between 0 and 1. The closer to 1 the better. Values towards zero indicate low performance. It can be either estimated for each particular class or at a global level.

Metrica offers 4 identical alternative functions that do the same job: i) `recall`, ii) `sensitivity`, iii) `TPR`, and iv) `hitrate`. However, consider when using `metrics_summary`, only the `recall` alternative is used.

The false negative rate (or false alarm, or fall-out) is the complement of the recall, representing the ratio between the number of false negatives (FN) to the actual number of positives (P). The `FNR` formula is:

$FNR = 1 - recall = 1 - TPR = \frac{FN}{P}$

The `fpr` is bounded between 0 and 1. The closer to 0 the better. Low performance is indicated with fpr > 0.5.

For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Ting K.M. (2017) Precision and Recall. *In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA.* doi:10.1007/9781489976871_659

Ting K.M. (2017). Sensitivity. *In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA.* doi:10.1007/9781489976871_751

Trevethan, R. (2017). *Sensitivity, Specificity, and Predictive Values: Foundations, Pliabilities, and Pitfalls* _ in Research and Practice. Front. Public Health 5:307_ doi:10.3389/fpubh.2017.00307

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100,
replace = TRUE), predictions = sample(c("True","False"), 100, replace = TRUE))

# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100,
replace = TRUE), predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE))

# Get recall estimate for two-class case at global level
recall(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get FNR estimate for two-class case at global level
FNR(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get recall estimate for each class for the multi-class case at global level
recall(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE,
```

```
atom = FALSE)

# Get recall estimate for the multi-class case at a class-level
recall(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE,
atom = TRUE)
```

RMAE *Relative Mean Absolute Error (RMAE)*

Description

It estimates the RMAE for a continuous predicted-observed dataset.

Usage

```
RMAE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The RMAE normalizes the Mean Absolute Error (MAE) by the mean of observations. The closer to zero the lower the prediction error. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
RMAE(obs = X, pred = Y)
```

RMLA *Root Mean Lack of Accuracy (RMLA)*

Description

It estimates the RMLA, the square root of the systematic error component to the Mean Squared Error (MSE), for a continuous predicted-observed dataset following Correndo et al. (2021).

Usage

```
RMLA(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The RMLA represents the systematic (bias) component of the MSE expressed into the original variable units. It is obtained via a symmetric decomposition of the MSE (invariant to predicted-observed orientation) using a symmetric regression line (SMA). The RMLA is equal to the square-root of the sum of systematic differences divided by the sample size (n). The greater the value the greater the bias of the predictions. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Correndo et al. (2021). Revisiting linear regression to test agreement in continuous predicted-observed datasets. *Agric. Syst. 192, 103194.* doi:10.1016/j.agsy.2021.103194

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
RMLA(obs = X, pred = Y)
```

RMLP *Root Mean Lack of Precision (RMLP)*

Description

It estimates the RMLP, the square root of the unsystematic error component to the Mean Squared Error (MSE), for a continuous predicted-observed dataset following Correndo et al. (2021).

Usage

```
RMLP(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The RMLP represents the unsystematic (random) component of the MSE expressed on the original variables units $\sqrt{MLP}$. It is obtained via a symmetric decomposition of the MSE (invariant to predicted-observed orientation) using a symmetric regression line (SMA). The RMLP is equal to the square-root of the sum of unsystematic differences divided by the sample size (n). The greater the value the greater the random noise of the predictions. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Correndo et al. (2021). Revisiting linear regression to test agreement in continuous predicted-observed datasets. *Agric. Syst. 192, 103194.* doi:10.1016/j.agsy.2021.103194

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
RMLP(obs = X, pred = Y)
```

RMSE *Root Mean Squared Error (RMSE)*

Description

It estimates the RMSE for a continuous predicted-observed dataset.

Usage

```
RMSE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The RMSE is one of the most widely used error metrics in literature to evaluate prediction performance. It measures general agreement, being sensitive to both lack of precision and lack of accuracy. It is simply the square root of the Mean Squared Error (MSE). Thus, it presents the same units as the variable of interest, facilitating the interpretation. It goes from 0 to infinity. The lower the value the better the prediction performance. Its counterpart is being very sensitive to outliers. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
RMSE(obs = X, pred = Y)
```

RRMSE *Relative Root Mean Squared Error (RMSE)*

Description

It estimates the RRMSE for a continuous predicted-observed dataset.

Usage

```
RRMSE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The RRMSE normalizes the Root Mean Squared Error (RMSE) by the mean of observations. It goes from 0 to infinity. The lower the better the prediction performance. In literature, it can be also found as NRMSE (normalized root mean squared error). However, here we use RRMSE since several other alternatives to "normalize" the RMSE exist (e.g., RSR, iqRMSE). For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
RRMSE(obs = X, pred = Y)
```

RSE *Relative Squared Error (RSE)*

Description

It estimates the RSE for a continuous predicted-observer dataset.

Usage

```
RSE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The RSE is the ratio between the residual sum of squares (RSS, error of predictions with respect to observations) and the total sum of squares (TSS, error of observations with respect to its mean). RSE is dimensionless, so it can be used to compared models with different units. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
RSE(obs = X, pred = Y)
```

RSR *Root Mean Standard Deviation Ratio (RSR)*

Description

It estimates the MSE normalized by the standard deviation of observed values following Moriasi et al. (2007).

Usage

```
RSR(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The RSR normalizes the Root Mean Squared Error (RMSE) using the standard deviation of observed values. It goes from an optimal value of 0 to infinity. Based on RSR, Moriasi et al. (2007) indicates performance ratings as: i) very-good (0-0.50), ii) good (0.50-0.60), iii) satisfactory (0.60-0.70), or iv) unsatisfactory (>0.70). For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Moriasi et al. (2007). Model Evaluation Guidelines for Systematic Quantification of Accuracy in Watershed Simulations. *Trans. ASABE 50, 885–900.* doi:10.13031/2013.23153

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
RSR(obs = X, pred = Y)
```

RSS *Residual Sum of Squares (RSS)*

Description

It estimates the RSS for a continuous predicted-observed dataset.

Usage

```
RSS(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The RSS is the sum of the squared differences between predictions and observations. It represents the base of many error metrics using squared scale such as the Mean Squared Error (MSE). For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
RSS(obs = X, pred = Y)
```

SB *Squared bias (SB)*

Description

It estimates the SB component of the Mean Squared Error (MSE) proposed by Kobayashi & Salam (2000).

Usage

```
SB(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The SB represents the additive bias component of the prediction error following Kobayashi & Salam (2000). It is in square units of the variable of interest, so it does not have a direct interpretation. The lower the value the less contribution to the MSE. However, it needs to be compared to MSE as its benchmark. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Kobayashi & Salam (2000). Comparing simulated and measured values using mean squared deviation and its components. *Agron. J. 92, 345–352.* doi:10.2134/agronj2000.922345x

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 9)
SB(obs = X, pred = Y)
```

scatter_plot *Scatter plot of predicted and observed values*

Description

It draws a scatter plot of predictions and observations with alternative axis orientation (P vs. O; O vs. P).

Usage

```
scatter_plot(
  data = NULL,
  obs,
  pred,
  orientation = "PO",
  print_metrics = FALSE,
  metrics_list = NULL,
  position_metrics = c(x = NULL, y = NULL),
  print_eq = TRUE,
  position_eq = c(x = NULL, y = NULL),
  eq_color = NULL,
  shape_type = NULL,
  shape_size = NULL,
  shape_color = NULL,
  shape_fill = NULL,
  regline_type = NULL,
  regline_size = NULL,
  regline_color = NULL,
  na.rm = TRUE
)
```

Arguments

data	(Optional) argument to call an existing data frame with the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
orientation	Argument of class string specifying the axis orientation, PO for predicted vs observed, and OP for observed vs predicted. Default is orientation = "PO".
print_metrics	boolean TRUE/FALSE to embed metrics in the plot. Default is FALSE.
metrics_list	vector or list of selected metrics to print on the plot.
position_metrics	vector or list with '(x,y)' coordinates to locate the metrics_table into the plot. Default : c(x = min(obs), y = 1.05*max(pred)).
print_eq	boolean TRUE/FALSE to embed metrics in the plot. Default is FALSE.

position_eq	vector or list with '(x,y)' coordinates to locate the SMA equation into the plot. Default : c(x = 0.70 max(x), y = 1.25*min(y)).
eq_color	string indicating the color of the SMA-regression text.
shape_type	integer indicating the shape type for the data points.
shape_size	number indicating the shape size for the data points.
shape_color	string indicating the shape color for the data points.
shape_fill	string indicating the shape fill for the data points.
regline_type	string or integer indicating the SMA-regression line-type.
regline_size	number indicating the SMA-regression linewidth.
regline_color	string indicating the SMA-regression line color.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

It creates a scatter plot of predicted vs. observed values. The plot also includes the 1:1 line (solid line) and the linear regression line (dashed line). By default, it places the observed on the x-axis and the predicted on the y-axis (orientation = "PO"). This can be inverted by changing the argument orientation = "OP". For more details, see online-documentation

Value

an object of class ggplot.

See Also

ggplot,geom_point,aes

Examples

```
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 10)
scatter_plot(obs = X, pre = Y)
```

SDSD *Squared difference between standard deviations (SDSD)*

Description

It estimates the SDSD component of the Mean Squared Error (MSE) proposed by Kobayashi & Salam (2000).

Usage

```
SDSD(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The SDSD represents the proportional bias component of the prediction error following Kobayashi & Salam (2000). It is in square units of the variable of interest, so it does not have a direct interpretation. The lower the value the less contribution to the MSE. However, it needs to be compared to MSE as its benchmark. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Kobayashi & Salam (2000). Comparing simulated and measured values using mean squared deviation and its components. *Agron. J. 92, 345–352.* doi:10.2134/agronj2000.922345x

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
SDSD(obs = X, pred = Y)
```

SMAPE *Symmetric Mean Absolute Percentage Error (SMAPE).*

Description

It estimates the SMAPE for a continuous predicted-observed dataset.

Usage

```
SMAPE(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The SMAPE (%) is a normalized, dimensionless, and bounded (0% to 200%). It is a modification of the MAPE where the denominator is half of the sum of absolute differences between observations and predictions. This modification solves the problem of MAPE of producing negative or undefined values. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Makridakis (1993). Accuracy measures: theoretical and practical concerns. *Int. J. Forecast. 9, 527-529.* doi:10.1016/01692070(93)900793

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
SMAPE(obs = X, pred = Y)
```

---

sorghum *Sorghum grain number*

Description

This example dataset is a set of APSIM simulations of sorghum grain number (x1000 grains per squared meter), which exhibits both low accuracy and low precision as it represents a model still under development. The experimental trials come from 6 site-years in 1 country (Australia). These data correspond to the latest, up-to-date, documentation and validation of version number 2020.03.27.4956. Data available at: doi:10.7910/DVN/EJS4M0. Further details can be found at the official APSIM Next Generation website: https://APSIMnextgeneration.netlify.app/modeldocumentation/

Usage

```
sorghum
```

Format

This data frame has 36 rows and the following 2 columns:

pred predicted values

obs observed values

Source

https://github.com/adriancorrendo/metrica

`specificity` *Specificity | Selectivity | True Negative Rate*

Description

`specificity` estimates the specificity (a.k.a. selectivity, or true negative rate -TNR-) for a nominal/categorical predicted-observed dataset.

`selectivity` alternative to `specificity()`.

`TNR` alternative to `specificity()`.

`FPR` estimates the false positive rate (a.k.a fall-out or false alarm) for a nominal/categorical predicted-observed dataset.

Usage

```
specificity(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)

selectivity(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
```

```
)

TNR(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)

FPR(
  data = NULL,
  obs,
  pred,
  atom = FALSE,
  pos_level = 2,
  tidy = FALSE,
  na.rm = TRUE
)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (character | factor).
pred	Vector with predicted values (character | factor).
atom	Logical operator (TRUE/FALSE) to decide if the estimate is made for each class (atom = TRUE) or at a global level (atom = FALSE); Default : FALSE. When dataset is "binomial" atom does not apply.
pos_level	Integer, for binary cases, indicating the order (1|2) of the level corresponding to the positive. Generally, the positive level is the second (2) since following an alpha-numeric order, the most common pairs are (Negative | Positive), (0 | 1), (FALSE | TRUE). Default : 2.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The specificity (or selectivity, or true negative rate-TNR-) is a non-normalized coefficient that represents the ratio between the correctly negative predicted cases (or true negative -TN- for binary cases) to the total number of actual observations not belonging to a given class (actual negatives -N- for binary cases).

For binomial cases, $specificity = \frac{TN}{N} = \frac{TN}{TN+FP}$

The specificity metric is bounded between 0 and 1. The closer to 1 the better. Values towards zero indicate low performance. For multinomial cases, it can be either estimated for each particular class or at a global level.

Metrica offers 3 identical alternative functions that do the same job: i) specificity, ii) selectivity, and iii) TNR. However, consider when using metrics_summary, only the specificity alternative is used.

The false positive rate (or false alarm, or fall-out) is the complement of the specificity, representing the ratio between the number of false positives (FP) to the actual number of negatives (N). The FPR formula is:

$FPR = 1 - specificity = 1 - TNR = \frac{FP}{N}$

The FPR is bounded between 0 and 1. The closer to 0 the better. Low performance is indicated with FPR > 0.5.

For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

References

Ting K.M. (2017) Sensitivity and Specificity. *In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA*. doi:10.1007/9780387301648_752

Trevethan, R. (2017). *Sensitivity, Specificity, and Predictive Values: Foundations, Pliabilities, and Pitfalls* _ in Research and Practice. Front. Public Health 5:307_ doi:10.3389/fpubh.2017.00307

Examples

```
set.seed(123)
# Two-class
binomial_case <- data.frame(labels = sample(c("True","False"), 100,
replace = TRUE), predictions = sample(c("True","False"), 100, replace = TRUE))
# Multi-class
multinomial_case <- data.frame(labels = sample(c("Red","Blue", "Green"), 100,
replace = TRUE), predictions = sample(c("Red","Blue", "Green"), 100, replace = TRUE)    )

# Get specificity and FPR estimates for two-class case
specificity(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)
FPR(data = binomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get specificity estimate for each class for the multi-class case
specificity(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)

# Get specificity estimate for the multi-class case at a global level
specificity(data = multinomial_case, obs = labels, pred = predictions, tidy = TRUE)
```

`tiles_plot` *Tiles plot of predicted and observed values*

Description

It draws a tiles plot of predictions and observations with alternative axis orientation (P vs. O; O vs. P).

Usage

```
tiles_plot(
  data = NULL,
  obs,
  pred,
  bins = 10,
  colors = c(low = NULL, high = NULL),
  orientation = "PO",
  print_metrics = FALSE,
  metrics_list = NULL,
  position_metrics = c(x = NULL, y = NULL),
  print_eq = TRUE,
  position_eq = c(x = NULL, y = NULL),
  eq_color = NULL,
  regline_type = NULL,
  regline_size = NULL,
  regline_color = NULL,
  na.rm = TRUE
)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`bins`	Argument of class numeric specifying the number of bins to create the tiles.
`colors`	Vector or list with two colors '(low, high)' to paint the density gradient.
`orientation`	Argument of class string specifying the axis orientation, PO for predicted vs observed, and OP for observed vs predicted. Default is orientation = "PO".
`print_metrics`	boolean TRUE/FALSE to embed metrics in the plot. Default is FALSE.
`metrics_list`	vector or list of selected metrics to print on the plot.
`position_metrics`	vector or list with '(x,y)' coordinates to locate the metrics_table into the plot. Default : c(x = min(obs), y = 1.05*max(pred)).
`print_eq`	boolean TRUE/FALSE to embed metrics in the plot. Default is FALSE.

position_eq	vector or list with '(x,y)' coordinates to locate the SMA equation into the plot. Default : c(x = 0.70 max(x), y = 1.25*min(y)).
eq_color	string indicating the color of the SMA-regression text.
regline_type	string or integer indicating the SMA-regression line-type.
regline_size	number indicating the SMA-regression line size.
regline_color	string indicating the SMA-regression line color.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

It creates a tiles plot of predicted vs. observed values. The plot also includes the 1:1 line (solid line) and the linear regression line (dashed line). By default, it places the observed on the x-axis and the predicted on the y-axis (orientation = "PO"). This can be inverted by changing the argument orientation = "OP". For more details, see online-documentation

Value

Object of class ggplot.

See Also

ggplot,geom_point,aes

Examples

```
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- rnorm(n = 100, mean = 0, sd = 10)
tiles_plot(obs = X, pred = Y)
```

TSS *Total Sum of Squares (TSS)*

Description

It estimates the TSS for observed vector.

Usage

```
TSS(data = NULL, obs, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The TSS sum of the squared differences between the observations and its mean. It is used as a reference error, for example, to estimate explained variance. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
TSS(obs = X)
```

Ub *Mean Bias Error Proportion (Ub)*

Description

It estimates the Ub component from the sum of squares decomposition described by Smith & Rose (1995).

Usage

```
Ub(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The Ub estimates the proportion of the total sum of squares related to the mean bias following the sum of squares decomposition suggested by Smith and Rose (1995) also known as Theil's partial inequalities. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Smith & Rose (1995). Model goodness-of-fit analysis using regression and related techniques. *Ecol. Model. 77, 49–64.* doi:10.1016/03043800(93)E0074D

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
Ub(obs = X, pred = Y)
```

Uc *Lack of Consistency (Uc)*

Description

It estimates the Uc component from the sum of squares decomposition described by Smith & Rose (1995).

Usage

```
Uc(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The Uc estimates the proportion of the total sum of squares related to the lack of consistency (proportional bias) following the sum of squares decomposition suggested by Smith and Rose (1995) also known as Theil's partial inequalities. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Smith & Rose (1995). Model goodness-of-fit analysis using regression and related techniques. *Ecol. Model. 77, 49–64.* doi:10.1016/03043800(93)E0074D

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
Uc(obs = X, pred = Y)
```

Ue *Lack of Consistency (Ue)*

Description

It estimates the Ue component from the sum of squares decomposition described by Smith & Rose (1995).

Usage

```
Ue(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`obs`	Vector with observed values (numeric).
`pred`	Vector with predicted values (numeric).
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE). The Ue estimates the proportion of the total sum of squares related to the random error (unsystematic error or variance) following the sum of squares decomposition suggested by Smith and Rose (1995) also known as Theil's partial inequalities. For the formula and more details, see online-documentation

References

Smith & Rose (1995). Model goodness-of-fit analysis using regression and related techniques. *Ecol. Model. 77, 49–64.* doi:10.1016/03043800(93)E0074D

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
Ue(obs = X, pred = Y)
```

---

uSD *Uncorrected Standard Deviation*

Description

It estimates the (uSD) of observed or predicted values.

Usage

```
uSD(data = NULL, x, tidy = FALSE, na.rm = TRUE)
```

Arguments

`data`	(Optional) argument to call an existing data frame containing the data.
`x`	Vector with numeric observed or predicted values.
`tidy`	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
`na.rm`	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The uSD is the sample, uncorrected standard deviation. The square root of the mean of sum of squared differences between vector values with respect to their mean. It is uncorrected because it is divided by the sample size (n), not n-1. For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
uSD(x = X)
```

---

var_u *Uncorrected Variance (var_u)*

Description

It estimates the var_u of observed or predicted values.

Usage

```
var_u(data = NULL, x, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
x	Vector with numeric elements.
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The var_u is the sample, uncorrected variance. It is calculated as the mean of sum of squared differences between values of an x and its mean, divided by the sample size (n). It is uncorrected because it is divided by n, and by not n-1 (traditional variance). For the formula and more details, see online-documentation

Value

an object of class numeric within a list (if tidy = FALSE) or within a data frame (if tidy = TRUE).

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
var_u(x = X)
```

wheat *Wheat grain nitrogen*

Description

This example dataset is a set of APSIM simulations of wheat grain N (grams per squared meter), which presents both high accuracy and high precision. The experimental trials come from 11 site-years in 4 countries (Australia, Ethiopia, New Zealand, and Turkey). These data correspond to the latest, up-to-date, documentation and validation of version number 2020.03.27.4956. Data available at: doi:10.7910/DVN/EJS4M0. Further details can be found at the official APSIM Next Generation website: https://APSIMnextgeneration.netlify.app/modeldocumentation/

Usage

```
wheat
```

Format

This data frame has 137 rows and the following 2 columns:

pred predicted values

obs observed values

Source

https://github.com/adriancorrendo/metrica

Xa *Accuracy Component (Xa) of CCC*

Description

It estimates the Xa component for the calculation of the Concordance Correlation Coefficient (CCC) following Lin (1989).

Usage

```
Xa(data = NULL, obs, pred, tidy = FALSE, na.rm = TRUE)
```

Arguments

data	(Optional) argument to call an existing data frame containing the data.
obs	Vector with observed values (numeric).
pred	Vector with predicted values (numeric).
tidy	Logical operator (TRUE/FALSE) to decide the type of return. TRUE returns a data.frame, FALSE returns a list; Default : FALSE.
na.rm	Logic argument to remove rows with missing values (NA). Default is na.rm = TRUE.

Details

The Xa measures accuracy of prediction. It goes from 0 (completely inaccurate) to 1 (perfectly accurate). It is used to adjust the precision measured by the correlation coefficient (r) in order to evaluate agreement through the CCC. For the formula and more details, see online-documentation

Value

an object of class `numeric` within a `list` (if tidy = FALSE) or within a `data frame` (if tidy = TRUE).

References

Lin (1989). A concordance correlation coefficient to evaluate reproducibility. *Biometrics 45 (1), 255–268.* doi:10.2307/2532051

Examples

```
set.seed(1)
X <- rnorm(n = 100, mean = 0, sd = 10)
Y <- X + rnorm(n=100, mean = 0, sd = 3)
Xa(obs = X, pred = Y)
```

Index