

# Package ‘rmetrics’

December 8, 2022

**Type** Package

**Title** A Collection of Metrics to Evaluate the Performance of Statistical and Machine Learning Models

**Version** 0.1.0

**Description** The package provides a common interface to a very large collection of metrics for the evaluation of machine learning models and statistical models.  
It is mainly designed as a backend for the revaluate package, but each function can of course be used on its own.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.2

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

## R topics documented:

calc_acc . . . . .	3
calc_acc_macro . . . . .	4
calc_af . . . . .	5
calc_aickin . . . . .	6
calc_aunp . . . . .	7
calc_aunu . . . . .	8
calc_aupr . . . . .	9
calc_auroc . . . . .	10
calc_b . . . . .	11
calc_bbs . . . . .	12
calc_bcd . . . . .	13
calc_bennett_s . . . . .	15
calc_cba . . . . .	15
calc_chisq . . . . .	16
calc_conditional_entropy . . . . .	17
calc_cramer_v . . . . .	18
calc_cross_entropy . . . . .	19
calc_csi_macro . . . . .	20
calc_dind . . . . .	21

calc_dor	22
calc_dp	23
calc_err	24
calc_f	25
calc_fdr	26
calc_fmi	27
calc_fnr	28
calc_fnr_macro	29
calc_fnr_micro	30
calc_for	31
calc_fpr	32
calc_fpr_macro	33
calc_fpr_micro	34
calc_f_macro	35
calc_gini	36
calc_gmean	37
calc_grey	38
calc_gwet_ac1	39
calc_hamming	40
calc_iba	41
calc_icsi	42
calc_informedness	43
calc_is	44
calc_jaccard	45
calc_jaccard_overall	46
calc_joint_entropy	47
calc_kalpha	47
calc_kappa	48
calc_kl_divergence	49
calc_lambda	50
calc_lift	52
calc_markedness	53
calc_mcc	54
calc_mcc_overall	55
calc_mutual_information	56
calc_net_benefit	56
calc_nir	57
calc_nlr	58
calc_npv	59
calc_oacc	60
calc_oc	61
calc_ooc	62
calc_op	63
calc_oracc	64
calc_pearson_c	65
calc_phi	65
calc_phisq	66
calc_plr	67
calc_ppv_macro	68
calc_precision	69
calc_prevalence	70
calc_q	71

<i>calc_acc</i>	3
-----------------	---

<i>calc_racc</i>	72
<i>calc_rand</i>	73
<i>calc_rci</i>	74
<i>calc_reference_entropy</i>	75
<i>calc_response_entropy</i>	76
<i>calc_rr</i>	77
<i>calc_sind</i>	78
<i>calc_tnr</i>	79
<i>calc_tnr_macro</i>	80
<i>calc_tnr_micro</i>	80
<i>calc_tpr</i>	81
<i>calc_tpr_macro</i>	82
<i>calc_zero_one_loss</i>	83

<b>Index</b>	<b>85</b>
--------------	-----------

---

<i>calc_acc</i>	<i>Calculate Accuracy (acc)</i>
-----------------	---------------------------------

---

## Description

Accuracy refers to the degree to which a model's predictions are correct. It is a measure of how well a model is able to make predictions that match the observed data. Accuracy is calculated by dividing the number of correct predictions made by the model by the total number of predictions made.

## Usage

```
calc_acc(...)

## Default S3 method:
calc_acc(tp, tn, fp, fn, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_acc(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_acc(data, prediction, reference, ci.type, ci.level, ...)
```

## Arguments

<i>...</i>	Additional arguments. Not used.
<i>tp</i>	Numeric, True Positives (TP).
<i>tn</i>	Numeric, True Negatives (TN).
<i>fp</i>	Numeric, False Positives (FP).
<i>fn</i>	Numeric, False Negatives (FN).
<i>ci.type</i>	Either FALSE if no confidence intervals are desired or one of "agresti.coull", "agresti-coull", "ac", "asymptotic", "normal", "wald", "clopper-pearson", "cp", "exact", "jeffreys", "bayes", and "wilson". If FALSE, overwrites <i>ci.level</i> .
<i>ci.level</i>	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.

tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Details

Accuracy is calculated as follows:

Accuracy = (Number of correct predictions) / (Total number of predictions)

For example, if a model makes 100 predictions and 75 of them are correct, the accuracy score would be  $75/100 = 0.75$ . This means that the model is able to correctly predict the outcome 75% of the time.

Overall, accuracy is a useful metric for evaluating the performance of a predictive model, but it should be used in conjunction with other metrics to get a complete picture of a model's performance, especially when class imbalance is present or a focus on either type 1 or type 2 errors is necessary.

## Methods (by class)

- `calc_acc(default)`:
- `calc_acc(table)`:
- `calc_acc(data.frame)`:

---

calc_acc_macro	<i>Calculate Accuracy Macro.</i>
----------------	----------------------------------

---

## Description

Calculate Accuracy Macro.

## Usage

```
calc_acc_macro(...)

## Default S3 method:
calc_acc_macro(tp, tn, n, ...)

## S3 method for class 'table'
calc_acc_macro(tbl, ...)

## S3 method for class 'data.frame'
calc_acc_macro(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
tn	Numeric vector of True Negatives (TN) by class.
n	Total number of observations.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_acc\_macro(default):
- calc\_acc\_macro(table):
- calc\_acc\_macro(data.frame):

calc\_af

*Calculate Adjusted F-score (af).***Description**

Calculate Adjusted F-score (af).

**Usage**

```
calc_af(...)

## Default S3 method:
calc_af(tp, fp, fn, tn, ...)

## S3 method for class 'table'
calc_af(tbl, ...)

## S3 method for class 'data.frame'
calc_af(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fp	Numeric, False Positives (FP).
fn	Numeric, False Negatives (FN).
tn	Numeric, True Negatives (TN).

tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Methods (by class)

- calc\_af(default):
- calc\_af(table):
- calc\_af(data.frame):

---

calc_aickin	<i>Calculate Aickin's Alpha.</i>
-------------	----------------------------------

---

### Description

Calculate Aickin's Alpha.

### Usage

```
calc_aickin(...)

## S3 method for class 'table'
calc_aickin(
  tbl,
  d = diag(1, nrow = nrow(tbl), ncol = ncol(tbl)),
  epsilon = 1e-07,
  ci.type = "aickin",
  ci.level = 0.95,
  maxiter = 1000,
  ...
)

## S3 method for class 'data.frame'
calc_aickin(
  data,
  prediction,
  reference,
  d = diag(1, nrow = nrow(tbl), ncol = ncol(tbl)),
  epsilon = 1e-07,
  ci.type = "aickin",
  ci.level = 0.95,
  maxiter = 1000,
  ...
)
```

**Arguments**

...	Additional arguments. Not used.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
d	The agreement matrix for tbl. Must have the same dimensions as tbl, with 1 to indicate agreeing scores, and 0 disagreeing scores. By default the diagonal is considered agreeing.
epsilon	Convergence criterion. The algorithm stops when two consecutive $\alpha$ estimates differ by less than epsilon. Default is 1e-7.
ci.type	Either FALSE if no confidence intervals are desired or 'aickin' if the default CI's by Aickin should be calculated. If FALSE overwrites ci.level. Default is to calculate the CI.
ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
maxiter	Integer or double. Maximum number of iterations to try until convergence. Default is 1000.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_aickin(table):
- calc\_aickin(data.frame):

**Author(s)**

Shamelessly stolen and only slightly adapted from Joseph, L. and Bélisle, P. <https://www.medicine.mcgill.ca/epidemiology/Joseph/PBelisle/Aickin-Alpha-Agreement-R.html>

**Source**

1. Aickin, M. (1990) Maximum Likelihood Estimation of Agreement in the Constant Predictive Probability Model, and Its Relation to Cohen's Kappa. Biometrics 46, 293-302.

---

calc_aunp	<i>Calculate overall AUC (aunp).</i>
-----------	--------------------------------------

---

**Description**

Calculate overall AUC (aunp).

**Usage**

```
calc_aunp(...)

## Default S3 method:
calc_aunp(tp, tn, fp, fn, ...)

## S3 method for class 'table'
calc_aunp(tbl, ...)

## S3 method for class 'data.frame'
calc_aunp(data, prediction = "prediction", reference = "reference", ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
tn	Numeric vector of True Negatives (TN) by class.
fp	Numeric vector of False Positives (FP) by class.
fn	Numeric vector of False Negatives (FN) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_aunp(default):
- calc\_aunp(table):
- calc\_aunp(data.frame):

---

calc_aunu	<i>Calculate average AUC (aunu) / AUC macro.</i>
-----------	--

---

**Description**

Calculate average AUC (aunu) / AUC macro.

**Usage**

```
calc_aunu(...)

## Default S3 method:
calc_aunu(tp, tn, fp, fn, ...)

## S3 method for class 'table'
calc_aunu(tbl, ...)

## S3 method for class 'data.frame'
calc_aunu(data, prediction = "prediction", reference = "reference", ...)
```



**Arguments**

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
tn	Numeric vector of True Negatives (TN) by class.
fp	Numeric vector of False Positives (FP) by class.
fn	Numeric vector of False Negatives (FN) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_aunu(default):
- calc\_aunu(table):
- calc\_aunu(data.frame):

---

calc_aupr	<i>Calculate Area under the PR curve (AUPR).</i>
-----------	--

---

**Description**

The area under the precision-recall curve (AUPRC) is a metric that is used to evaluate the performance of a binary classification model. It is calculated by measuring the area under the curve generated by plotting the precision and recall of a model as functions of a probability threshold.

**Usage**

```
calc_aupr(...)

## Default S3 method:
calc_aupr(tp, fp, fn, ...)

## S3 method for class 'table'
calc_aupr(tbl, ...)

## S3 method for class 'data.frame'
calc_aupr(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fp	Numeric, False Positives (FP).
fn	Numeric, False Negatives (FN).

tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Details

Precision and recall are two metrics that are commonly used to evaluate the performance of a binary classification model. Precision is the ratio of true positive predictions to the total number of positive predictions made by the model, while recall is the ratio of true positive predictions to the total number of actual positive samples.

The AUPRC is calculated by first generating a series of precision-recall pairs at different probability thresholds. This is done by varying the threshold used to classify a sample as positive or negative, and calculating the precision and recall at each threshold. These precision-recall pairs are then plotted on a graph, with precision on the y-axis and recall on the x-axis. The AUPRC is then calculated by measuring the area under this curve.

The AUPRC is a useful metric because it allows for a more comprehensive evaluation of a model's performance. Unlike the F1-score, which is a single number that represents the harmonic mean of precision and recall, the AUPRC takes into account the entire precision-recall curve and provides a more complete picture of a model's performance.

## Methods (by class)

- `calc_aupr(default)`:
- `calc_aupr(table)`:
- `calc_aupr(data.frame)`:

---

calc_auroc	<i>Calculate Area under the ROC curve for each class (auroc).</i>
------------	---

---

## Description

The area under the curve (AUC) is a metric that is used to evaluate the performance of a binary classification model. It is calculated by measuring the area under the curve generated by plotting the true positive rate (TPR) and false positive rate (FPR) as functions of a probability threshold.

## Usage

```
calc_auroc(...)

## Default S3 method:
calc_auroc(tn, fp, tp, fn, ...)

## S3 method for class 'table'
calc_auroc(tbl, ...)

## S3 method for class 'data.frame'
calc_auroc(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Details**

The TPR and FPR are two metrics that are commonly used to evaluate the performance of a binary classification model. The TPR is the ratio of true positive predictions to the total number of actual positive samples, while the FPR is the ratio of false positive predictions to the total number of actual negative samples.

The AUC can be calculated by first generating a series of TPR-FPR pairs at different probability thresholds. This is done by varying the threshold used to classify a sample as positive or negative, and calculating the TPR and FPR at each threshold. These pairs are then plotted on a graph, with the TPR on the y-axis and the FPR on the x-axis. The AUC is then calculated by measuring the area under this curve.

The AUC is a useful metric because it allows for a more comprehensive evaluation of a model's performance. The AUC takes into account the entire TPR-FPR curve and thereby provides a more complete picture of a model's performance.

**Methods (by class)**

- `calc_auroc(default):`
- `calc_auroc(table):`
- `calc_auroc(data.frame):`

---

calc_b	<i>Calculate Bangdiwala's B.</i>
--------	----------------------------------

---

**Description**

Calculate Bangdiwala's B.

**Usage**

```
calc_b(...)

## Default S3 method:
calc_b(tp, fp, fn, ...)

## S3 method for class 'table'
calc_b(tbl, ...)

## S3 method for class 'data.frame'
calc_b(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fp	Numeric vector of False Positives (FP) by class.
fn	Numeric vector of False Negatives (FN) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_b(default):
- calc\_b(table):
- calc\_b(data.frame):

---

calc\_bbs

---

*Calculate Braun-Blanquet similarity.*


---

**Description**

The Braun-Blanquet similarity is a measure of the degree to which two sets of observations are similar. It is commonly used in ecology to compare the composition of two different plant communities, but it can also be applied to other types of data sets.

**Usage**

```
calc_bbs(...)

## Default S3 method:
calc_bbs(tp, ppos, pos, ...)

## S3 method for class 'table'
```

```
calc_bbs(tbl, ...)

## S3 method for class 'data.frame'
calc_bbs(data, prediction, reference, ...)
```

### Arguments

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
ppos	Number of positives in prediction vector (= TP + FP)
pos	Number of positives in reference.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Details

To calculate the Braun-Blanquet similarity, the following steps are followed:

- For each observation in the first set of data, count the number of times it occurs.
- For each observation in the second set of data, count the number of times it occurs.
- For each observation that occurs in both sets of data, calculate the minimum of the two counts and add this value to the Braun-Blanquet similarity score.
- Divide the Braun-Blanquet similarity score by the total number of observations in both sets of data.

The resulting value is a measure of the degree to which the two sets of data are similar, with values closer to 1 indicating greater similarity and values closer to 0 indicating less similarity.

### Methods (by class)

- calc\_bbs(default):
- calc\_bbs(table):
- calc\_bbs(data.frame):

---

calc\_bcd

---

*Calculate Bray-Curtis dissimilarity.*


---

### Description

The Bray-Curtis dissimilarity is a measure of the degree to which two sets of observations are different. It is commonly used in ecology to compare the composition of two different plant communities, but it can also be applied to other types of data sets.

**Usage**

```
calc_bcd(...)

## Default S3 method:
calc_bcd(ppos, pos, n, ...)

## S3 method for class 'table'
calc_bcd(tbl, ...)

## S3 method for class 'data.frame'
calc_bcd(data, prediction, reference, ...)
```

**Arguments**

<code>...</code>	Additional arguments. Not used.
<code>ppos</code>	Number of positives in prediction vector (= TP + FP)
<code>pos</code>	Number of positives in reference.
<code>n</code>	Total number of observations.
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in <i>rmetrics</i> will generate incorrect results.
<code>data</code>	A <i>data.frame</i> containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in <i>data</i> that contains the predictions.
<code>reference</code>	Character. The name of the variable in <i>data</i> that contains the reference values.

**Details**

To calculate the Bray-Curtis dissimilarity, the following steps are followed:

- For each observation in the first set of data, count the number of times it occurs.
- For each observation in the second set of data, count the number of times it occurs.
- For each observation that occurs in both sets of data, calculate the absolute difference between the two counts and add this value to the Bray- Curtis dissimilarity score.
- Divide the Bray-Curtis dissimilarity score by the total number of observations in both sets of data.

The resulting value is a measure of the degree to which the two sets of data are different, with values closer to 1 indicating greater dissimilarity and values closer to 0 indicating less dissimilarity.

**Methods (by class)**

- `calc_bcd(default)`:
- `calc_bcd(table)`:
- `calc_bcd(data.frame)`:

---

calc_bennett_s	<i>Calculate Bennett's S.</i>
----------------	-------------------------------

---

**Description**

Calculate Bennett's S.

**Usage**

```
calc_bennett_s(...)

## S3 method for class 'table'
calc_bennett_s(tbl, ...)

## S3 method for class 'data.frame'
calc_bennett_s(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_bennett\_s(table):
- calc\_bennett\_s(data.frame):

---

calc_cba	<i>Calculate Class Balance Accuracy.</i>
----------	--

---

**Description**

Calculate Class Balance Accuracy.

**Usage**

```
calc_cba(...)

## Default S3 method:
calc_cba(tp, fp, fn, ...)

## S3 method for class 'table'
calc_cba(tbl, ...)

## S3 method for class 'data.frame'
calc_cba(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fp	Numeric vector of False Positives (FP) by class.
fn	Numeric vector of False Negatives (FN) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_cba(default):
- calc\_cba(table):
- calc\_cba(data.frame):

---

calc_chisq	<i>Calculate Chi-squared.</i>
------------	-------------------------------

---

**Description**

A simple wrapper over `stats::chisq.test`

**Usage**

```
calc_chisq(...)

## S3 method for class 'table'
calc_chisq(tbl, ...)

## S3 method for class 'data.frame'
calc_chisq(data, prediction, reference, ...)
```

**Arguments**

...	Additional parameters passed on to <code>chisq.test</code> .
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Value**

A named vector with the test statistic, degrees of freedom and p value of the test.



**Methods (by class)**

- `calc_chisq(table)`:
- `calc_chisq(data.frame)`:

---

`calc_conditional_entropy`
*Calculate conditional entropy.*


---

**Description**

Calculate conditional entropy.

**Usage**

```
calc_conditional_entropy(...)

## S3 method for class 'data.frame'
calc_conditional_entropy(
  data,
  prediction = "prediction",
  reference = "reference",
  ...
)

## S3 method for class 'table'
calc_conditional_entropy(tbl, ...)
```

**Arguments**

<code>...</code>	Additional arguments. Not used. Computes the average conditional entropy between two vectors.
<code>data</code>	A data.frame containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in data that contains the predictions.
<code>reference</code>	Character. The name of the variable in data that contains the reference values.
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.

**Methods (by class)**

- `calc_conditional_entropy(data.frame)`:
- `calc_conditional_entropy(table)`:

---

calc_cramer_v	<i>Calculate Cramer's V.</i>
---------------	------------------------------

---

## Description

Calculate Cramer's V.

## Usage

```
calc_cramer_v(...)

## S3 method for class 'table'
calc_cramer_v(
  tbl,
  ci.type = c("ncchisq", "ncchisqadj", "fisher", "fisheradj"),
  ci.level = 0.95,
  bias.correct = FALSE,
  ...
)

## S3 method for class 'data.frame'
calc_cramer_v(
  data,
  prediction = "prediction",
  reference = "reference",
  ci.type = c("ncchisq", "ncchisqadj", "fisher", "fisheradj"),
  ci.level = 0.95,
  bias.correct = FALSE,
  ...
)
```

## Arguments

...	Additional arguments passed on to <code>stats::chisq.test</code> . Not used.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in <code>rmetrics</code> will generate incorrect results.
ci.type	FALSE if no ci is requested or one out of "ncchisq" (using noncentral chisquare), "ncchisqadj", "fisher" (using fisher z transformation), "fisheradj" (using fisher z transformation and bias correction).
ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
bias.correct	Should a bias correction be applied? FALSE by default.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Details**

A Cramer's V between 0 and 0.3 is considered as weak, 0.3-0.7 as medium and  $> 0.7$  as strong. Equivalent to Tschuprow's T in square tables.

**Value**

A numeric vector with the three elements 'v', 'll' and 'ul'. If no CI is requested, 'll' and 'ul' are NA.

**Methods (by class)**

- `calc_cramer_v(table)`:
- `calc_cramer_v(data.frame)`:

**Author(s)**

This implementation is based on code by Andri Signorell [andri@signorell.net](mailto:andri@signorell.net) and Michael Smithson [michael.smithson@anu.edu.au](mailto:michael.smithson@anu.edu.au) for the package DescTools with only minor changes.

**References**

Cramer, H. (1946) Mathematical Methods of Statistics. Princeton University Press  
 Agresti, A. (2002) Categorical Data Analysis. John Wiley & Sons

---

calc_cross_entropy	<i>Calculate Cross entropy.</i>
--------------------	---------------------------------

---

**Description**

Calculate Cross entropy.

**Usage**

```
calc_cross_entropy(...)

## Default S3 method:
calc_cross_entropy(tp, fp, fn, n, ...)

## S3 method for class 'table'
calc_cross_entropy(tbl, ...)

## S3 method for class 'data.frame'
calc_cross_entropy(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fp	Numeric vector of False Positives (FP) by class.
fn	Numeric vector of False Negatives (FN) by class.

n	Total number of observations.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Methods (by class)

- calc\_cross\_entropy(default):
- calc\_cross\_entropy(table):
- calc\_cross\_entropy(data.frame):

---

calc\_csi\_macro

*Calculate classification Success Index Macro.*

---

### Description

Calculate classification Success Index Macro.

### Usage

```
calc_csi_macro(...)

## Default S3 method:
calc_csi_macro(tp, fn, fp, ...)

## S3 method for class 'table'
calc_csi_macro(tbl, ...)

## S3 method for class 'data.frame'
calc_csi_macro(data, prediction, reference, ...)
```

### Arguments

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fn	Numeric vector of False Negatives (FN) by class.
fp	Numeric vector of False Positives (FP) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- `calc_csi_macro(default):`
- `calc_csi_macro(table):`
- `calc_csi_macro(data.frame):`

---

calc_dind	<i>Calculate Distance Index.</i>
-----------	----------------------------------

---

**Description**

Calculate Distance Index.

**Usage**

```
calc_dind(...)

## Default S3 method:
calc_dind(tn, fp, tp, fn, ...)

## S3 method for class 'table'
calc_dind(tbl, ...)

## S3 method for class 'data.frame'
calc_dind(data, prediction, reference, ...)
```

**Arguments**

<code>...</code>	Additional arguments. Not used.
<code>tn</code>	Numeric, True Negatives (TN).
<code>fp</code>	Numeric, False Positives (FP).
<code>tp</code>	Numeric, True Positives (TP).
<code>fn</code>	Numeric, False Negatives (FN).
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in <code>rmetrics</code> will generate incorrect results.
<code>data</code>	A <code>data.frame</code> containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in <code>data</code> that contains the predictions.
<code>reference</code>	Character. The name of the variable in <code>data</code> that contains the reference values.

**Methods (by class)**

- `calc_dind(default):`
- `calc_dind(table):`
- `calc_dind(data.frame):`

---

calc_dor	<i>Calculate Diagnostic odds ratio</i>
----------	--

---

## Description

The diagnostic odds ratio (DOR) is a metric that is used to evaluate the performance of a diagnostic test. It is calculated as the ratio of the odds of a positive test result in people with the disease to the odds of a positive test result in people without the disease.

## Usage

```
calc_dor(...)

## Default S3 method:
calc_dor(tp, fn, tn, fp, ...)

## S3 method for class 'table'
calc_dor(tbl, ...)

## S3 method for class 'data.frame'
calc_dor(data, prediction, reference, ...)
```

## Arguments

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Details

To calculate the DOR, the following formula is used:

$$\text{DOR} = (\text{True positive rate} / \text{False positive rate}) / (\text{False negative rate} / \text{True negative rate})$$

where the true positive rate is the ratio of true positive predictions to the total number of actual positive samples, the false positive rate is the ratio of false positive predictions to the total number of actual negative samples, the false negative rate is the ratio of false negative predictions to the total number of actual positive samples, and the true negative rate is the ratio of true negative predictions to the total number of actual negative samples.

The resulting value is a measure of the effectiveness of a diagnostic test, with values closer to 1 indicating a weaker test and values closer to infinity indicating a stronger test.

**Methods (by class)**

- `calc_dp(default):`
- `calc_dp(table):`
- `calc_dp(data.frame):`

---

calc_dp	<i>Calculate Discriminant Power.</i>
---------	--------------------------------------

---

**Description**

Calculate Discriminant Power.

**Usage**

```
calc_dp(...)

## Default S3 method:
calc_dp(tn, fp, tp, fn, ...)

## S3 method for class 'table'
calc_dp(tbl, ...)

## S3 method for class 'data.frame'
calc_dp(data, prediction, reference, ...)
```

**Arguments**

<code>...</code>	Additional arguments. Not used.
<code>tn</code>	Numeric, True Negatives (TN).
<code>fp</code>	Numeric, False Positives (FP).
<code>tp</code>	Numeric, True Positives (TP).
<code>fn</code>	Numeric, False Negatives (FN).
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in <code>rmetrics</code> will generate incorrect results.
<code>data</code>	A data.frame containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in data that contains the predictions.
<code>reference</code>	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- `calc_dp(default):`
- `calc_dp(table):`
- `calc_dp(data.frame):`

calc\_err

*Calculate Error Rate.***Description**

The error rate of a classification model is a measure of the proportion of incorrect predictions made by the model. It is calculated as the number of incorrect predictions divided by the total number of predictions made by the model.

**Usage**

```
calc_err(...)

## Default S3 method:
calc_err(tp, tn, fp, fn, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_err(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_err(data, prediction, reference, ci.type, ci.level, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
fn	Numeric, False Negatives (FN).
ci.type	Either FALSE if no confidence intervals are desired or one of "agresti.coull", "agresti-coull", "ac", "asymptotic", "normal", "wald", "clopper-pearson", "cp", "exact", "jeffreys", "bayes", and "wilson". If FALSE, overwrites ci.level.
ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Details**

To calculate the error rate of a classification model, the following formula is used:

Error rate = (Number of incorrect predictions) / (Total number of predictions)

The resulting value is a proportion, with values closer to 0 indicating a better-performing model and values closer to 1 indicating a worse-performing model.



**Methods (by class)**

- `calc_err(default):`
- `calc_err(table):`
- `calc_err(data.frame):`

---

<code>calc_f</code>	<i>Calculate (adjusted) F-Score</i>
---------------------	-------------------------------------

---

**Description**

The F-score, also known as the F1-score or F-measure, is a metric that combines precision and recall into a single score. This implementation allows you to set the parameter beta to control the relative weight of precision and recall.

**Usage**

```
calc_f(...)

## Default S3 method:
calc_f(tp, fp, fn, beta, ...)

## S3 method for class 'table'
calc_f(tbl, beta, ...)

## S3 method for class 'data.frame'
calc_f(data, prediction, reference, beta, ...)
```

**Arguments**

<code>...</code>	Additional arguments. Not used.
<code>tp</code>	Numeric, True Positives (TP).
<code>fp</code>	Numeric, False Positives (FP).
<code>fn</code>	Numeric, False Negatives (FN).
<code>beta</code>	beta coefficient
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
<code>data</code>	A data.frame containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in data that contains the predictions.
<code>reference</code>	Character. The name of the variable in data that contains the reference values.

## Details

To calculate the adjusted F-score, the following formula is used:

$$\text{Adjusted F-score} = (1 + \beta^2) \cdot (\text{Precision} \cdot \text{Recall}) / (\beta^2 \cdot \text{Precision} + \text{Recall})$$

where  $\beta$  is a parameter that controls the relative weight of precision and recall. For  $\beta = 1$ , the adjusted F-score is equivalent to the F1-score. For  $\beta < 1$ , precision is given more weight, and for  $\beta > 1$ , recall is given more weight.

Overall, the adjusted F-score is a useful metric for evaluating the performance of a predictive model, particularly when the precision and recall are calculated using a sample of the data. It allows for a more balanced evaluation of a model's performance by taking into account the number of data points used to calculate the precision and recall.

## Methods (by class)

- `calc_f(default)`:
- `calc_f(table)`:
- `calc_f(data.frame)`:

---

<code>calc_fdr</code>	<i>Calculate False Discovery Rate</i>
-----------------------	---------------------------------------

---

## Description

The false discovery rate (FDR) is a measure of the proportion of false positive results among all positive results in a statistical test. It is calculated as the number of false positives divided by the total number of positive results.

## Usage

```
calc_fdr(...)

## Default S3 method:
calc_fdr(fp, tp, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_fdr(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_fdr(data, prediction, reference, ci.type, ci.level, ...)
```

## Arguments

<code>...</code>	Additional arguments. Not used.
<code>fp</code>	Numeric, False Positives (FP).
<code>tp</code>	Numeric, True Positives (TP).
<code>ci.type</code>	Either FALSE if no confidence intervals are desired or one of "agresti.coull", "agresti-coull", "ac", "asymptotic", "normal", "wald", "clopper-pearson", "cp", "exact", "jeffreys", "bayes", and "wilson". If FALSE, overwrites <code>ci.level</code> .

ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Details

To calculate the FDR, the following formula is used:

$$\text{FDR} = (\text{Number of false positives}) / (\text{Total number of positive results})$$

The resulting value is a proportion, with values closer to 0 indicating a more reliable statistical test and values closer to 1 indicating a less reliable test.

### Methods (by class)

- calc\_fdr(default):
- calc\_fdr(table):
- calc\_fdr(data.frame):

---

calc_fmi	<i>Calculate Fowlkes–Mallows Index.</i>
----------	---

---

### Description

Calculate Fowlkes–Mallows Index.

### Usage

```
calc_fmi(...)

## S3 method for class 'table'
calc_fmi(tbl, ...)

## S3 method for class 'data.frame'
calc_fmi(data, prediction, reference, ...)
```

### Arguments

...	Additional arguments. Not used. The Fowlkes–Mallows index is the geometric mean of precision (PPV) and recall (TPR). It is generally used to compare the results of two clustering algorithms. It ranges from 0 to 1, with 1 indicating perfect classification.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- `calc_fmi(table)`:
- `calc_fmi(data.frame)`:

**References**

Fowlkes, E. B.; Mallows, C. L. (1 September 1983). "A Method for Comparing Two Hierarchical Clusterings". *Journal of the American Statistical Association*. 78 (383): 553. doi:10.2307/2288117

---

calc_fnr	<i>Calculate False Negative Rate</i>
----------	--------------------------------------

---

**Description**

The false negative rate is a measure of the proportion of false negative results in a binary classification problem. It is calculated as the number of false negative predictions divided by the total number of actual positive samples.

**Usage**

```
calc_fnr(...)

## Default S3 method:
calc_fnr(fn, tp, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_fnr(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_fnr(data, prediction, reference, ci.type, ci.level, ...)
```

**Arguments**

<code>...</code>	Additional arguments. Not used.
<code>fn</code>	Numeric, False Negatives (FN).
<code>tp</code>	Numeric, True Positives (TP).
<code>ci.type</code>	Either FALSE if no confidence intervals are desired or one of "agresti.coull", "agresti-coull", "ac", "asymptotic", "normal", "wald", "clopper-pearson", "cp", "exact", "jeffreys", "bayes", and "wilson". If FALSE, overwrites ci.level.
<code>ci.level</code>	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
<code>data</code>	A data.frame containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in data that contains the predictions.
<code>reference</code>	Character. The name of the variable in data that contains the reference values.

## Details

To calculate the false negative rate, the following formula is used:

False negative rate = (Number of false negative predictions) / (Total number of actual positive samples)

The resulting value is a proportion, with values closer to 0 indicating a better-performing model and values closer to 1 indicating a worse- performing model.

Overall, the false negative rate is a useful metric for evaluating the performance of a classification model. It provides a way to compare the ability of different models to accurately predict the positive class, and can be used to identify models that have a high degree of accuracy for the positive class.

## Methods (by class)

- `calc_fnr(default)`:
- `calc_fnr(table)`:
- `calc_fnr(data.frame)`:

---

<code>calc_fnr_macro</code>	<i>Calculate FNR macro.</i>
-----------------------------	-----------------------------

---

## Description

Calculate FNR macro.

## Usage

```
calc_fnr_macro(...)

## Default S3 method:
calc_fnr_macro(fn, tp, ...)

## S3 method for class 'table'
calc_fnr_macro(tbl, ...)

## S3 method for class 'data.frame'
calc_fnr_macro(data, prediction, reference, ...)
```

## Arguments

<code>...</code>	Additional arguments. Not used.
<code>fn</code>	Numeric vector of False Negatives (FN) by class.
<code>tp</code>	Numeric vector of True Positives (TP) by class.
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in <code>rmetrics</code> will generate incorrect results.
<code>data</code>	A <code>data.frame</code> containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in <code>data</code> that contains the predictions.
<code>reference</code>	Character. The name of the variable in <code>data</code> that contains the reference values.

**Methods (by class)**

- calc\_fnr\_macro(default):
- calc\_fnr\_macro(table):
- calc\_fnr\_macro(data.frame):

---

calc_fnr_micro	<i>Calculate FNR micro.</i>
----------------	-----------------------------

---

**Description**

Calculate FNR micro.

**Usage**

```
calc_fnr_micro(...)

## Default S3 method:
calc_fnr_micro(otp, n, ...)

## S3 method for class 'table'
calc_fnr_micro(tbl, ...)

## S3 method for class 'data.frame'
calc_fnr_micro(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
otp	Overall True Positives (OTP).
n	Total number of observations.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_fnr\_micro(default):
- calc\_fnr\_micro(table):
- calc\_fnr\_micro(data.frame):

---

calc_for	<i>Calculate False Omission Rate</i>
----------	--------------------------------------

---

## Description

The false omission rate (FOR) is a measure of the proportion of false negative results among all negative results in a statistical test. It is calculated as the number of false negatives divided by the total number of negative results.

## Usage

```
calc_for(...)

## Default S3 method:
calc_for(fn, tn, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_for(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_for(data, prediction, reference, ci.type, ci.level, ...)
```

## Arguments

...	Additional arguments. Not used.
fn	Numeric, False Negatives (FN).
tn	Numeric, True Negatives (TN).
ci.type	Either FALSE if no confidence intervals are desired or one of "agresti.coull", "agresti-coull", "ac", "asymptotic", "normal", "wald", "clopper-pearson", "cp", "exact", "jeffreys", "bayes", and "wilson". If FALSE, overwrites ci.level.
ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Details

To calculate the FOR, the following formula is used:

$$\text{FOR} = (\text{Number of false negatives}) / (\text{Total number of negative results})$$

The resulting value is a proportion, with values closer to 0 indicating a more reliable statistical test and values closer to 1 indicating a less reliable test.

The FOR is a useful metric to identify tests that are more likely to accurately identify true negative results.

**Methods (by class)**

- `calc_for(default):`
- `calc_for(table):`
- `calc_for(data.frame):`

---

`calc_fpr`*Calculate False Positive Rate*

---

**Description**

The false positive rate (FPR) is a measure of the proportion of false positive results in a binary classification problem. It is calculated as the number of false positive predictions divided by the total number of actual negative samples.

**Usage**

```
calc_fpr(...)

## Default S3 method:
calc_fpr(fp, tn, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_fpr(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_fpr(data, prediction, reference, ci.type, ci.level, ...)
```

**Arguments**

<code>...</code>	Additional arguments. Not used.
<code>fp</code>	Numeric, False Positives (FP).
<code>tn</code>	Numeric, True Negatives (TN).
<code>ci.type</code>	Either FALSE if no confidence intervals are desired or one of "agresti.coull", "agresti-coull", "ac", "asymptotic", "normal", "wald", "clopper-pearson", "cp", "exact", "jeffreys", "bayes", and "wilson". If FALSE, overwrites <code>ci.level</code> .
<code>ci.level</code>	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in <code>rmetrics</code> will generate incorrect results.
<code>data</code>	A <code>data.frame</code> containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in <code>data</code> that contains the predictions.
<code>reference</code>	Character. The name of the variable in <code>data</code> that contains the reference values.



## Details

To calculate the FPR, the following formula is used:

False positive rate = (Number of false positive predictions) / (Total number of actual negative samples)

The resulting value is a proportion, with values closer to 0 indicating a better-performing model and values closer to 1 indicating a worse- performing model.

The FPR is a useful metric to compare the ability of different models to accurately predict the negative class, and can be used to identify models that have a high degree of accuracy for the negative class.

## Methods (by class)

- `calc_fpr(default)`:
- `calc_fpr(table)`:
- `calc_fpr(data.frame)`:

---

<code>calc_fpr_macro</code>	<i>Calculate FPR Macro.</i>
-----------------------------	-----------------------------

---

## Description

Calculate FPR Macro.

## Usage

```
calc_fpr_macro(...)

## Default S3 method:
calc_fpr_macro(fp, tn, ...)

## S3 method for class 'table'
calc_fpr_macro(tbl, ...)

## S3 method for class 'data.frame'
calc_fpr_macro(data, prediction, reference, ...)
```

## Arguments

<code>...</code>	Additional arguments. Not used.
<code>fp</code>	Numeric vector of False Positives (FP) by class.
<code>tn</code>	Numeric vector of True Negatives (TN) by class.
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in <code>rmetrics</code> will generate incorrect results.
<code>data</code>	A <code>data.frame</code> containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in <code>data</code> that contains the predictions.
<code>reference</code>	Character. The name of the variable in <code>data</code> that contains the reference values.

**Methods (by class)**

- calc\_fpr\_macro(default):
- calc\_fpr\_macro(table):
- calc\_fpr\_macro(data.frame):

---

calc_fpr_micro	<i>Calculate FPR Micro.</i>
----------------	-----------------------------

---

**Description**

Calculate FPR Micro.

**Usage**

```
calc_fpr_micro(...)

## Default S3 method:
calc_fpr_micro(tn, fp, ...)

## S3 method for class 'table'
calc_fpr_micro(tbl, ...)

## S3 method for class 'data.frame'
calc_fpr_micro(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tn	Numeric vector of True Negatives (TN) by class.
fp	Numeric vector of False Positives (FP) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_fpr\_micro(default):
- calc\_fpr\_micro(table):
- calc\_fpr\_micro(data.frame):

---

calc_f_macro	<i>Calculate F macro.</i>
--------------	---------------------------

---

## Description

Calculate F macro.

## Usage

```
calc_f_macro(...)  
  
## Default S3 method:  
calc_f_macro(tp, fp, fn, beta = 1, ...)  
  
## S3 method for class 'table'  
calc_f_macro(tbl, beta = 1, ...)  
  
## S3 method for class 'data.frame'  
calc_f_macro(data, prediction, reference, beta = 1, ...)
```

## Arguments

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fp	Numeric vector of False Positives (FP) by class.
fn	Numeric vector of False Negatives (FN) by class.
beta	Scaling factor. 1 by default for the F1-Score.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_f\_macro(default):
- calc\_f\_macro(table):
- calc\_f\_macro(data.frame):

---

calc_gini	<i>Calculate Gini Index</i>
-----------	-----------------------------

---

**Description**

Calculate Gini Index

**Usage**

```
calc_gini(...)

## Default S3 method:
calc_gini(tn, fp, tp, fn, ...)

## S3 method for class 'table'
calc_gini(tbl, ...)

## S3 method for class 'data.frame'
calc_gini(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_gini(default):
- calc\_gini(table):
- calc\_gini(data.frame):

calc\_gmean

*Calculate (adjusted) Geometric mean of TPR and TNR.***Description**

The geometric mean is a measure of central tendency that is calculated by taking the n-th root of the product of n numbers. In the context of binary classification, the true positive rate (TPR) and true negative rate (TNR) are two common metrics used to evaluate the performance of a model. The TPR is the ratio of true positive predictions to the total number of positive samples, and the TNR is the ratio of true negative predictions to the total number of negative samples.

**Usage**

```
calc_gmean(...)

## Default S3 method:
calc_gmean(tn, fp, tp, fn, adjust = FALSE, ...)

## S3 method for class 'table'
calc_gmean(tbl, adjust = FALSE, ...)

## S3 method for class 'data.frame'
calc_gmean(data, prediction, reference, adjust = FALSE, ...)
```

**Arguments**

...	Additional arguments. Not used.
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
adjust	Should the GM be adjusted for the proportion of negatives? FALSE by default.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Details**

The geometric mean of TPR and TNR can be adjusted for the proportion of negatives by setting `adjust = TRUE`.

**Methods (by class)**

- `calc_gmean(default):`
- `calc_gmean(table):`
- `calc_gmean(data.frame):`

calc\_grey

*Calculate Size of Grey Area***Description**

Calculate Size of Grey Area

**Usage**

```
calc_grey(...)

## Default S3 method:
calc_grey(ind, n, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_grey(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_grey(data, prediction, reference, ci.type, ci.level, ...)
```

**Arguments**

...	Additional arguments. Not used.
ind	Number of indeterminate results.
n	Total number of observations.
ci.type	Either FALSE if no confidence intervals are desired or one of "agresti.coull", "agresti-coull", "ac", "asymptotic", "normal", "wald", "clopper-pearson", "cp", "exact", "jeffreys", "bayes", and "wilson". If FALSE, overwrites ci.level.
ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_grey(default):
- calc\_grey(table):
- calc\_grey(data.frame):

---

calc_gwet_ac1	<i>Calculate Gwet AC1.</i>
---------------	----------------------------

---

### Description

Calculate Gwet AC1.

### Usage

```
calc_gwet_ac1(...)

## Default S3 method:
calc_gwet_ac1(tp, fp, fn, n, ...)

## S3 method for class 'table'
calc_gwet_ac1(tbl, ...)

## S3 method for class 'data.frame'
calc_gwet_ac1(data, prediction, reference, ...)
```

### Arguments

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fp	Numeric vector of False Positives (FP) by class.
fn	Numeric vector of False Negatives (FN) by class.
n	Total number of observations.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Methods (by class)

- calc\_gwet\_ac1(default):
- calc\_gwet\_ac1(table):
- calc\_gwet\_ac1(data.frame):

---

calc_hamming	<i>Calculate Hamming Loss.</i>
--------------	--------------------------------

---

## Description

Calculate Hamming Loss.

## Usage

```
calc_hamming(...)

## Default S3 method:
calc_hamming(otp, n, ...)

## S3 method for class 'table'
calc_hamming(tbl, ...)

## S3 method for class 'data.frame'
calc_hamming(data, prediction, reference, ...)
```

## Arguments

...	Additional arguments. Not used.
otp	Overall True Positives (OTP).
n	Total number of observations.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_hamming(default):
- calc\_hamming(table):
- calc\_hamming(data.frame):



---

calc_iba	<i>Calculate Index of Balanced Accuracy.</i>
----------	--

---

## Description

Calculate Index of Balanced Accuracy.

## Usage

```
calc_iba(...)

## Default S3 method:
calc_iba(tn, fp, tp, fn, alpha = 1, ...)

## S3 method for class 'table'
calc_iba(tbl, alpha = 1, ...)

## S3 method for class 'data.frame'
calc_iba(data, prediction, reference, alpha = 1, ...)
```

## Arguments

...	Additional arguments. Not used.
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
alpha	Weight for TPR - TNR. By default 1.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_iba(default):
- calc\_iba(table):
- calc\_iba(data.frame):

---

calc_icsi	<i>Calculate Individual classification success index.</i>
-----------	---

---

## Description

Calculate Individual classification success index.

## Usage

```
calc_icsi(...)

## Default S3 method:
calc_icsi(tp, fn, fp, ...)

## S3 method for class 'table'
calc_icsi(tbl, ...)

## S3 method for class 'data.frame'
calc_icsi(data, prediction, reference, ...)
```

## Arguments

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
fp	Numeric, False Positives (FP).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_icsi(default):
- calc\_icsi(table):
- calc\_icsi(data.frame):

---

calc_informedness	<i>Calculate Informedness</i>
-------------------	-------------------------------

---

## Description

Calculate Informedness

## Usage

```
calc_informedness(...)

## Default S3 method:
calc_informedness(tp, fn, tn, fp, ...)

## S3 method for class 'table'
calc_informedness(tbl, ...)

## S3 method for class 'data.frame'
calc_informedness(data, prediction, reference, ...)
```

## Arguments

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_informedness(default):
- calc\_informedness(table):
- calc\_informedness(data.frame):

---

calc_is	<i>Calculate Information score.</i>
---------	-------------------------------------

---

### Description

Calculate Information score.

### Usage

```
calc_is(...)

## Default S3 method:
calc_is(tp, fp, fn, n, ...)

## S3 method for class 'table'
calc_is(tbl, ...)

## S3 method for class 'data.frame'
calc_is(data, prediction, reference, ...)
```

### Arguments

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fp	Numeric, False Positives (FP).
fn	Numeric, False Negatives (FN).
n	Total number of observations.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Methods (by class)

- calc\_is(default):
- calc\_is(table):
- calc\_is(data.frame):

---

calc_jaccard	<i>Calculate Jaccard index</i>
--------------	--------------------------------

---

## Description

Calculate Jaccard index

## Usage

```
calc_jaccard(...)  
  
## Default S3 method:  
calc_jaccard(tp, fn, fp, ...)  
  
## S3 method for class 'table'  
calc_jaccard(tbl, ...)  
  
## S3 method for class 'data.frame'  
calc_jaccard(data, prediction, reference, ...)
```

## Arguments

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
fp	Numeric, False Positives (FP).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_jaccard(default):
- calc\_jaccard(table):
- calc\_jaccard(data.frame):

---

calc\_jaccard\_overall    *Calculate Mean overall Jaccard index.*

---

### Description

Calculate Mean overall Jaccard index.

### Usage

```
calc_jaccard_overall(...)

## Default S3 method:
calc_jaccard_overall(tp, fn, fp, ...)

## S3 method for class 'table'
calc_jaccard_overall(tbl, ...)

## S3 method for class 'data.frame'
calc_jaccard_overall(
  data,
  prediction = "prediction",
  reference = "reference",
  ...
)
```

### Arguments

...	Additional arguments. Not used. The overall Jaccard index is calculated as the mean Jaccard index over all classes in data.
tp	Numeric vector of True Positives (TP) by class.
fn	Numeric vector of False Negatives (FN) by class.
fp	Numeric vector of False Positives (FP) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Methods (by class)

- calc\_jaccard\_overall(default):
- calc\_jaccard\_overall(table):
- calc\_jaccard\_overall(data.frame):

---

calc_joint_entropy	<i>Calculate Joint entropy.</i>
--------------------	---------------------------------

---

### Description

Calculate Joint entropy.

### Usage

```
calc_joint_entropy(...)

## S3 method for class 'table'
calc_joint_entropy(tbl, ...)

## S3 method for class 'data.frame'
calc_joint_entropy(data, prediction, reference, ...)
```

### Arguments

...	Additional arguments. Not used.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Methods (by class)

- calc\_joint\_entropy(table):
- calc\_joint\_entropy(data.frame):

---

calc_kalpha	<i>Calculate Unweighted Krippendorff's Alpha.</i>
-------------	---

---

### Description

Krippendorff's alpha is a statistical measure of the agreement among a set of raters who assign ratings to a set of items. It is commonly used in the field of content analysis, where researchers use it to assess the reliability of their coding schemes. It can be thought of as a generalization of the concept of inter-rater reliability, which is a measure of how well two or more raters agree on their ratings of a set of items. Krippendorff's alpha allows for the calculation of reliability even when there are more than two raters, and it can also be used with ordinal or continuous data, whereas inter-rater reliability is typically only used with binary or nominal data.

**Usage**

```
calc_kalpha(...)

## S3 method for class 'table'
calc_kalpha(tbl, unbiased = TRUE, ...)

## S3 method for class 'data.frame'
calc_kalpha(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
unbiased	TRUE/FALSE. Should unbiased overall random accuracy be used?
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Details**

The alpha value ranges from 0 to 1, where 0 indicates no agreement among the raters and 1 indicates perfect agreement. An alpha value of 0.7 or higher is generally considered to indicate good agreement among the raters.

**Methods (by class)**

- calc\_kalpha(table):
- calc\_kalpha(data.frame):

---

calc_kappa	<i>Calculate Kappa, unbiased Kappa or Kappa no Prevalence.</i>
------------	--

---

**Description**

Calculate Kappa, unbiased Kappa or Kappa no Prevalence.

**Usage**

```
calc_kappa(...)

## S3 method for class 'table'
calc_kappa(tbl, unbiased = FALSE, prev = TRUE, ...)

## S3 method for class 'data.frame'
calc_kappa(
  data,
  prediction = "prediction",
```



```

    reference = "reference",
    unbiased = FALSE,
    prev = TRUE,
    ...
  )

```

### Arguments

...	Additional arguments. Not used.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
unbiased	Logical, should 'normal' or unbiased overall random accuracy be used.
prev	TRUE for Kappa and unbiased Kappa, FALSE for Kappa no prevalence.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Methods (by class)

- calc\_kappa(table):
- calc\_kappa(data.frame):

---

calc_kl_divergence	<i>Calculate Kullback-Leibler Divergence.</i>
--------------------	---

---

### Description

Calculate Kullback-Leibler Divergence.

### Usage

```

calc_kl_divergence(...)

## Default S3 method:
calc_kl_divergence(tp, fp, fn, n, epsilon = 1e-06, ...)

## S3 method for class 'table'
calc_kl_divergence(tbl, epsilon = 1e-06, ...)

## S3 method for class 'data.frame'
calc_kl_divergence(data, prediction, reference, epsilon = 1e-06, ...)

```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fp	Numeric vector of False Positives (FP) by class.
fn	Numeric vector of False Negatives (FN) by class.
n	Total number of observations.
epsilon	Continuity correction for zero cells. By default 0.000001.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_kl\_divergence(default):
- calc\_kl\_divergence(table):
- calc\_kl\_divergence(data.frame):

---

calc_lambda	<i>Calculate Goodman Kruskal Lambda.</i>
-------------	--

---

**Description**

Calculate Goodman Kruskal Lambda.

**Usage**

```
calc_lambda(...)

## S3 method for class 'table'
calc_lambda(
  tbl,
  direction = c("symmetric", "row", "column"),
  ci.type = FALSE,
  ci.level = 0.95,
  ...
)

## S3 method for class 'data.frame'
calc_lambda(
  data,
  prediction = "prediction",
  reference = "reference",
  direction = c("symmetric", "row", "column"),
  ci.type = FALSE,
  ci.level = 0.95,
  ...
)
```

## Arguments

...	Additional arguments. Not used.
	Calculates symmetric and asymmetric (lambda A and lambda B) Goodman Kruskal lambda and their confidence intervals. Lambda measures the proportional reduction in error in cross tabulation analysis. It can be used to gauge the strength of association between two nominal variables. It can be interpreted as the probable improvement in predicting the reference given knowledge of the predictions.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
direction	Character, either "symmetric", "row" or "column". "row" corresponds to Lambda B and "column" to Lambda A.
ci.type	FALSE if no ci is requested or "normal" for normal approximation CIs
ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Details

This implementation is based on code by Andri Signorell [andri@signorell.net](mailto:andri@signorell.net), Antti Arppe [antti.arppe@helsinki.fi](mailto:antti.arppe@helsinki.fi) and Nanina Anderegg (confidence interval symmetric lambda) for the package DescTools.

## Value

A numeric vector with the three elements 'lambda', 'll' and 'ul'. If no CI is requested, 'll' and 'ul' are NA.

## Methods (by class)

- `calc_lambda(table):`
- `calc_lambda(data.frame):`

## References

Agresti, A. (2002) Categorical Data Analysis. John Wiley & Sons  
 Goodman, L. A., Kruskal W. H. (1979) Measures of Association for Cross Classifications. New York: Springer-Verlag (contains articles appearing in J. Amer. Statist. Assoc. in 1954, 1959, 1963, 1972).  
 Liebetrau, A. M. (1983) Measures of Association, Sage University Papers Series on Quantitative Applications in the Social Sciences, 07-004. Newbury Park, CA: Sage, pp. 17–24

---

calc_lift	<i>Calculate Lift Score</i>
-----------	-----------------------------

---

### Description

Calculate Lift Score

### Usage

```
calc_lift(...)

## Default S3 method:
calc_lift(tp, fp, pos, neg, ...)

## S3 method for class 'table'
calc_lift(tbl, ...)

## S3 method for class 'data.frame'
calc_lift(data, prediction, reference, ...)
```

### Arguments

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fp	Numeric, False Positives (FP).
pos	Number of positives in reference.
neg	Number of negatives in reference.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Methods (by class)

- calc\_lift(default):
- calc\_lift(table):
- calc\_lift(data.frame):

---

calc_markedness	<i>Calculate Markedness.</i>
-----------------	------------------------------

---

## Description

Calculate Markedness.

## Usage

```
calc_markedness(...)  
  
## Default S3 method:  
calc_markedness(tp, fn, tn, fp, ...)  
  
## S3 method for class 'table'  
calc_markedness(tbl, ...)  
  
## S3 method for class 'data.frame'  
calc_markedness(data, prediction, reference, ...)
```

## Arguments

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_markedness(default):
- calc\_markedness(table):
- calc\_markedness(data.frame):

calc\_mcc

*Calculate Matthews Correlation Coefficient***Description**

Calculate Matthews Correlation Coefficient

**Usage**

```
calc_mcc(...)

## Default S3 method:
calc_mcc(tp, tn, fp, fn, ...)

## S3 method for class 'table'
calc_mcc(tbl, ...)

## S3 method for class 'data.frame'
calc_mcc(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
fn	Numeric, False Negatives (FN).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_mcc(default):
- calc\_mcc(table):
- calc\_mcc(data.frame):

---

calc_mcc_overall	<i>Calculate Overall Matthews Correlation Coefficient.</i>
------------------	--

---

## Description

Calculate Overall Matthews Correlation Coefficient.

## Usage

```
calc_mcc_overall(...)  
  
## Default S3 method:  
calc_mcc_overall(tp, fp, fn, ...)  
  
## S3 method for class 'table'  
calc_mcc_overall(tbl, ...)  
  
## S3 method for class 'data.frame'  
calc_mcc_overall(data, prediction, reference, ...)
```

## Arguments

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fp	Numeric vector of False Positives (FP) by class.
fn	Numeric vector of False Negatives (FN) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_mcc\_overall(default):
- calc\_mcc\_overall(table):
- calc\_mcc\_overall(data.frame):

---

calc_mutual_information	<i>Calculate Mutual information.</i>
-------------------------	--------------------------------------

---

### Description

Calculate Mutual information.

### Usage

```
calc_mutual_information(...)

## S3 method for class 'table'
calc_mutual_information(tbl, ...)

## S3 method for class 'data.frame'
calc_mutual_information(data, prediction, reference, ...)
```

### Arguments

...	Additional arguments. Not used.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Methods (by class)

- calc\_mutual\_information(table):
- calc\_mutual\_information(data.frame):

---

calc_net_benefit	<i>Calculate Net Benefit.</i>
------------------	-------------------------------

---

### Description

Calculate Net Benefit.



**Usage**

```
calc_net_benefit(...)

## Default S3 method:
calc_net_benefit(tp, fp, n, weight = 1, ...)

## S3 method for class 'table'
calc_net_benefit(tbl, weight = 1, ...)

## S3 method for class 'data.frame'
calc_net_benefit(data, prediction, reference, weight = 1, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fp	Numeric, False Positives (FP).
n	Total number of observations.
weight	The weight for FP in comparison to TP. By default 1.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_net\_benefit(default):
- calc\_net\_benefit(table):
- calc\_net\_benefit(data.frame):

---

calc_nir	<i>Calculate No information Rate.</i>
----------	---------------------------------------

---

**Description**

Calculate No information Rate.

**Usage**

```
calc_nir(...)

## Default S3 method:
calc_nir(tp, fn, n, ...)

## S3 method for class 'table'
calc_nir(tbl, ...)

## S3 method for class 'data.frame'
calc_nir(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fn	Numeric vector of False Negatives (FN) by class.
n	Total number of observations.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_nlr(default):
- calc\_nlr(table):
- calc\_nlr(data.frame):

---

calc_nlr	<i>Calculate Negative Likelihood Ratio</i>
----------	--

---

**Description**

Calculate Negative Likelihood Ratio

**Usage**

```
calc_nlr(...)

## Default S3 method:
calc_nlr(tp, fn, fp, tn, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_nlr(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_nlr(data, prediction, reference, ci.type, ci.level, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
fp	Numeric, False Positives (FP).
tn	Numeric, True Negatives (TN).
ci.type	Either FALSE if no confidence intervals are desired or 'koopman'. If FALSE overwrites ci.level.

ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Methods (by class)

- calc\_nlr(default):
- calc\_nlr(table):
- calc\_nlr(data.frame):

### Source

Koopman, PAR (1984) Confidence intervals for the ratio of two binomial proportions. Biometrics; 513-517.

---

calc_npv	<i>Calculate Negative Predictive Value</i>
----------	--

---

### Description

Calculate Negative Predictive Value

### Usage

```
calc_npv(...)

## Default S3 method:
calc_npv(tn, fn, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_npv(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_npv(data, prediction, reference, ci.type, ci.level, ...)
```

### Arguments

...	Additional arguments. Not used.
tn	Numeric, True Negatives (TN).
fn	Numeric, False Negatives (FN).
ci.type	Either FALSE if no confidence intervals are desired or one of "agresti.coull", "agresti-coull", "ac", "asymptotic", "normal", "wald", "clopper-pearson", "cp", "exact", "jeffreys", "bayes", and "wilson". If FALSE, overwrites ci.level.

ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Methods (by class)

- calc\_npv(default):
- calc\_npv(table):
- calc\_npv(data.frame):

---

calc_oacc	<i>Calculate Calculate Overall Accuracy.</i>
-----------	--

---

### Description

Calculate Calculate Overall Accuracy.

### Usage

```
calc_oacc(...)

## Default S3 method:
calc_oacc(otp, n, ...)

## S3 method for class 'table'
calc_oacc(tbl, ...)

## S3 method for class 'data.frame'
calc_oacc(data, prediction, reference, ...)
```

### Arguments

...	Additional arguments. Not used. The proportion of overall true positives, regardless of class. Identical to micro-averaging TPR.
otp	Overall True Positives (OTP).
n	Total number of observations.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- `calc_oacc(default):`
- `calc_oacc(table):`
- `calc_oacc(data.frame):`

---

calc_oc	<i>Calculate Overlap Coefficient</i>
---------	--------------------------------------

---

**Description**

Calculate Overlap Coefficient

**Usage**

```
calc_oc(...)

## Default S3 method:
calc_oc(tp, fp, fn, ...)

## S3 method for class 'table'
calc_oc(tbl, ...)

## S3 method for class 'data.frame'
calc_oc(data, prediction, reference, ...)
```

**Arguments**

<code>...</code>	Additional arguments. Not used.
<code>tp</code>	Numeric, True Positives (TP).
<code>fp</code>	Numeric, False Positives (FP).
<code>fn</code>	Numeric, False Negatives (FN).
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in <code>rmetrics</code> will generate incorrect results.
<code>data</code>	A <code>data.frame</code> containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in <code>data</code> that contains the predictions.
<code>reference</code>	Character. The name of the variable in <code>data</code> that contains the reference values.

**Methods (by class)**

- `calc_oc(default):`
- `calc_oc(table):`
- `calc_oc(data.frame):`

---

calc_ooc	<i>Calculate Otsuka-Ochiai Coefficient</i>
----------	--

---

## Description

Calculate Otsuka-Ochiai Coefficient

## Usage

```
calc_ooc(...)

## Default S3 method:
calc_ooc(tp, fp, fn, ...)

## S3 method for class 'table'
calc_ooc(tbl, ...)

## S3 method for class 'data.frame'
calc_ooc(data, prediction, reference, ...)
```

## Arguments

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fp	Numeric, False Positives (FP).
fn	Numeric, False Negatives (FN).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_ooc(default):
- calc\_ooc(table):
- calc\_ooc(data.frame):

---

calc_op	<i>Calculate Optimized Precision.</i>
---------	---------------------------------------

---

**Description**

Calculate Optimized Precision.

**Usage**

```
calc_op(...)  
  
## Default S3 method:  
calc_op(tn, fp, tp, fn, ...)  
  
## S3 method for class 'table'  
calc_op(tbl, ...)  
  
## S3 method for class 'data.frame'  
calc_op(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_op(default):
- calc\_op(table):
- calc\_op(data.frame):

---

 calc\_oracc

---

*Calculate Calculate (Unbiased) Overall Random Accuracy.*


---

## Description

Calculate Calculate (Unbiased) Overall Random Accuracy.

## Usage

```
calc_oracc(...)

## Default S3 method:
calc_oracc(tp, fp, fn, n, unbiased = FALSE, ...)

## S3 method for class 'table'
calc_oracc(tbl, unbiased = FALSE, ...)

## S3 method for class 'data.frame'
calc_oracc(data, prediction, reference, unbiased = FALSE, ...)
```

## Arguments

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fp	Numeric vector of False Positives (FP) by class.
fn	Numeric vector of False Negatives (FN) by class.
n	Total number of observations.
unbiased	TRUE/FALSE, should unbiased random accuracy be returned? FALSE by default.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_oracc(default):
- calc\_oracc(table):
- calc\_oracc(data.frame):



---

calc_pearson_c	<i>Calculate Pearson's C.</i>
----------------	-------------------------------

---

**Description**

Calculate Pearson's C.

**Usage**

```
calc_pearson_c(...)

## S3 method for class 'table'
calc_pearson_c(tbl, ...)

## S3 method for class 'data.frame'
calc_pearson_c(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments passed on to <code>stats::chisq.test</code> .
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in <code>rmetrics</code> will generate incorrect results.
data	A <code>data.frame</code> containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- `calc_pearson_c(table):`
- `calc_pearson_c(data.frame):`

---

calc_phi	<i>Calculate Phi Coefficient</i>
----------	----------------------------------

---

**Description**

Calculate Phi Coefficient

**Usage**

```
calc_phi(...)

## Default S3 method:
calc_phi(tp, tn, fp, fn, ...)

## S3 method for class 'table'
calc_phi(tbl, ...)

## S3 method for class 'data.frame'
calc_phi(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
fn	Numeric, False Negatives (FN).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- `calc_phi(default)`:
- `calc_phi(table)`:
- `calc_phi(data.frame)`:

---

calc_phisq	<i>Calculate Phi-squared.</i>
------------	-------------------------------

---

**Description**

Calculate Phi-squared.

**Usage**

```
calc_phisq(...)

## S3 method for class 'table'
calc_phisq(tbl, ...)

## S3 method for class 'data.frame'
calc_phisq(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments passed on to <code>stats::chisq.test</code> .
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- `calc_phisq(table)`:
- `calc_phisq(data.frame)`:

calc\_plr

*Calculate Positive Likelihood Ratio***Description**

Calculate Positive Likelihood Ratio

**Usage**

```
calc_plr(...)

## Default S3 method:
calc_plr(tp, fn, fp, tn, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_plr(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_plr(data, prediction, reference, ci.type, ci.level, ...)
```

**Arguments**

<code>...</code>	Additional arguments. Not used.
<code>tp</code>	Numeric, True Positives (TP).
<code>fn</code>	Numeric, False Negatives (FN).
<code>fp</code>	Numeric, False Positives (FP).
<code>tn</code>	Numeric, True Negatives (TN).
<code>ci.type</code>	Either FALSE if no confidence intervals are desired or 'koopman'. If FALSE overwrites ci.level.
<code>ci.level</code>	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
<code>data</code>	A data.frame containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in data that contains the predictions.
<code>reference</code>	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- `calc_plr(default)`:
- `calc_plr(table)`:
- `calc_plr(data.frame)`:

**Source**

Koopman, PAR (1984) Confidence intervals for the ratio of two binomial proportions. Biometrics; 513-517.

---

calc_ppv_macro	<i>Calculate Precision (PPV) Macro.</i>
----------------	---

---

**Description**

Calculate Precision (PPV) Macro.

**Usage**

```
calc_ppv_macro(...)

## Default S3 method:
calc_ppv_macro(tp, fp, ...)

## S3 method for class 'table'
calc_ppv_macro(tbl, ...)

## S3 method for class 'data.frame'
calc_ppv_macro(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fp	Numeric vector of False Positives (FP) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_ppv\_macro(default):
- calc\_ppv\_macro(table):
- calc\_ppv\_macro(data.frame):

---

calc_precision	<i>Calculate Precision</i>
----------------	----------------------------

---

**Description**

Calculate Precision

**Usage**

```
calc_precision(...)

## Default S3 method:
calc_precision(tp, fp, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_precision(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_precision(data, prediction, reference, ci.type, ci.level, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fp	Numeric, False Positives (FP).
ci.type	Either FALSE if no confidence intervals are desired or one of "agresti.coull", "agresti-coull", "ac", "asymptotic", "normal", "wald", "clopper-pearson", "cp", "exact", "jeffreys", "bayes", and "wilson". If FALSE, overwrites ci.level.
ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_precision(default):
- calc\_precision(table):
- calc\_precision(data.frame):

---

calc_prevalence	<i>Calculate Sample Prevalence</i>
-----------------	------------------------------------

---

## Description

Calculate Sample Prevalence

## Usage

```
calc_prevalence(...)

## Default S3 method:
calc_prevalence(pos, neg, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_prevalence(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_prevalence(data, prediction, reference, ci.type, ci.level, ...)
```

## Arguments

...	Additional arguments. Not used.
pos	Number of positives in reference.
neg	Number of negatives in reference.
ci.type	Either FALSE if no confidence intervals are desired or one of "agresti.coull", "agresti-coull", "ac", "asymptotic", "normal", "wald", "clopper-pearson", "cp", "exact", "jeffreys", "bayes", and "wilson". If FALSE, overwrites ci.level.
ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_prevalence(default):
- calc\_prevalence(table):
- calc\_prevalence(data.frame):

---

calc_q	<i>Calculate Yule's Q.</i>
--------	----------------------------

---

**Description**

Calculate Yule's Q.

**Usage**

```
calc_q(...)  
  
## Default S3 method:  
calc_q(tp, tn, fp, fn, ...)  
  
## S3 method for class 'table'  
calc_q(tbl, ...)  
  
## S3 method for class 'data.frame'  
calc_q(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
fn	Numeric, False Negatives (FN).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_q(default):
- calc\_q(table):
- calc\_q(data.frame):

---

calc_racc	<i>Calculate (Unbiased) Random Accuracy</i>
-----------	---

---

## Description

Calculate (Unbiased) Random Accuracy

## Usage

```
calc_racc(...)

## Default S3 method:
calc_racc(tp, fp, fn, n, unbiased = FALSE, ...)

## S3 method for class 'table'
calc_racc(tbl, unbiased = FALSE, ...)

## S3 method for class 'data.frame'
calc_racc(data, prediction, reference, unbiased = FALSE, ...)
```

## Arguments

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fp	Numeric, False Positives (FP).
fn	Numeric, False Negatives (FN).
n	Total number of observations.
unbiased	Should unbiased random accuracy be calculated? FALSE by default.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_racc(default):
- calc\_racc(table):
- calc\_racc(data.frame):



---

calc_rand	<i>Calculate Rand Index.</i>
-----------	------------------------------

---

**Description**

Calculate Rand Index.

**Usage**

```
calc_rand(...)

## S3 method for class 'table'
calc_rand(tbl, adjust = FALSE, ...)

## S3 method for class 'data.frame'
calc_rand(
  data,
  prediction = "prediction",
  reference = "reference",
  adjust = FALSE,
  ...
)
```

**Arguments**

...	Additional arguments. Not used. The Rand index ranges between 0 and 1 and can be used to measure the similarity between two categorical vectors. It is commonly used to evaluate the similarity between clustering outcomes.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
adjust	Should the index be rescaled to take into account that random chance will cause some objects to occupy the same clusters, so that the Rand Index can never be zero? FALSE by default.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_rand(table):
- calc\_rand(data.frame):

**Author(s)**

Adapted from the implementation for Rand Index and Adjusted Rand index in fossil, written by Matthew Vavrek.

## References

Rand, W.M. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* 66: 846–850. Hubert, L. and Arabie, P. 1985. Comparing partitions. *Journal of Classification*. 2: 193–218.

---

calc_rci	<i>Calculate Relative Classifier Information.</i>
----------	---

---

## Description

Calculate Relative Classifier Information.

## Usage

```
calc_rci(...)

## S3 method for class 'table'
calc_rci(tbl, ...)

## S3 method for class 'data.frame'
calc_rci(data, prediction = "prediction", reference = "reference", ...)
```

## Arguments

...	Additional arguments. Not used.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_rci(table):
- calc\_rci(data.frame):

---

`calc_reference_entropy`*Calculate Reference Entropy.*

---

### Description

Calculate Reference Entropy.

### Usage

```
calc_reference_entropy(...)  
  
## Default S3 method:  
calc_reference_entropy(tp, fn, n, ...)  
  
## S3 method for class 'table'  
calc_reference_entropy(tbl, ...)  
  
## S3 method for class 'data.frame'  
calc_reference_entropy(data, prediction, reference, ...)
```

### Arguments

<code>...</code>	Additional arguments. Not used.
<code>tp</code>	Numeric vector of True Positives (TP) by class.
<code>fn</code>	Numeric vector of False Negatives (FN) by class.
<code>n</code>	Total number of observations.
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in <code>rmetrics</code> will generate incorrect results.
<code>data</code>	A <code>data.frame</code> containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in <code>data</code> that contains the predictions.
<code>reference</code>	Character. The name of the variable in <code>data</code> that contains the reference values.

### Methods (by class)

- `calc_reference_entropy(default):`
- `calc_reference_entropy(table):`
- `calc_reference_entropy(data.frame):`

---

calc\_response\_entropy *Calculate Response Entropy.*

---

## Description

Calculate Response Entropy.

## Usage

```
calc_response_entropy(...)

## Default S3 method:
calc_response_entropy(tp, fp, n, ...)

## S3 method for class 'table'
calc_response_entropy(tbl, ...)

## S3 method for class 'data.frame'
calc_response_entropy(data, prediction, reference, ...)
```

## Arguments

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fp	Numeric vector of False Positives (FP) by class.
n	Total number of observations.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_response\_entropy(default):
- calc\_response\_entropy(table):
- calc\_response\_entropy(data.frame):

---

calc_rr	<i>Calculate Global Performance Index.</i>
---------	--

---

## Description

Calculate Global Performance Index.

## Usage

```
calc_rr(...)  
  
## Default S3 method:  
calc_rr(tp, fp, ...)  
  
## S3 method for class 'table'  
calc_rr(tbl, ...)  
  
## S3 method for class 'data.frame'  
calc_rr(data, prediction, reference, ...)
```

## Arguments

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fp	Numeric vector of False Positives (FP) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_rr(default):
- calc\_rr(table):
- calc\_rr(data.frame):

---

calc_sind	<i>Calculate Similarity Index</i>
-----------	-----------------------------------

---

**Description**

Calculate Similarity Index

**Usage**

```
calc_sind(...)

## Default S3 method:
calc_sind(tn, fp, tp, fn, ...)

## S3 method for class 'table'
calc_sind(tbl, ...)

## S3 method for class 'data.frame'
calc_sind(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_sind(default):
- calc\_sind(table):
- calc\_sind(data.frame):

---

calc_tnr	<i>Calculate Specificity (TNR)</i>
----------	------------------------------------

---

## Description

Calculate Specificity (TNR)

## Usage

```
calc_tnr(...)

## Default S3 method:
calc_tnr(tn, fp, ci.type, ci.level, ...)

## S3 method for class 'table'
calc_tnr(tbl, ci.type, ci.level, ...)

## S3 method for class 'data.frame'
calc_tnr(data, prediction, reference, ci.type, ci.level, ...)
```

## Arguments

...	Additional arguments. Not used.
tn	Numeric, True Negatives (TN).
fp	Numeric, False Positives (FP).
ci.type	Either FALSE if no confidence intervals are desired or one of "agresti.coull", "agresti-coull", "ac", "asymptotic", "normal", "wald", "clopper-pearson", "cp", "exact", "jeffreys", "bayes", and "wilson". If FALSE, overwrites ci.level.
ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

## Methods (by class)

- calc\_tnr(default):
- calc\_tnr(table):
- calc\_tnr(data.frame):

---

calc_tnr_macro	<i>Calculate TNR Macro.</i>
----------------	-----------------------------

---

### Description

Calculate TNR Macro.

### Usage

```
calc_tnr_macro(...)

## Default S3 method:
calc_tnr_macro(tn, fp, ...)

## S3 method for class 'table'
calc_tnr_macro(tbl, ...)

## S3 method for class 'data.frame'
calc_tnr_macro(data, prediction, reference, ...)
```

### Arguments

...	Additional arguments. Not used.
tn	Numeric vector of True Negatives (TN) by class.
fp	Numeric vector of False Positives (FP) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Methods (by class)

- calc\_tnr\_macro(default):
- calc\_tnr\_macro(table):
- calc\_tnr\_macro(data.frame):

---

calc_tnr_micro	<i>Calculate TNR Micro.</i>
----------------	-----------------------------

---

### Description

Calculate TNR Micro.



**Usage**

```
calc_tnr_micro(...)

## Default S3 method:
calc_tnr_micro(tn, fp, ...)

## S3 method for class 'table'
calc_tnr_micro(tbl, ...)

## S3 method for class 'data.frame'
calc_tnr_micro(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tn	Numeric vector of True Negatives (TN) by class.
fp	Numeric vector of False Positives (FP) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_tnr\_micro(default):
- calc\_tnr\_micro(table):
- calc\_tnr\_micro(data.frame):

---

calc_tpr	<i>Calculate Sensitivity (TPR)</i>
----------	------------------------------------

---

**Description**

Sensitivity is the ability of a test or measurement to accurately detect what it is supposed to detect. In the context of medical tests, sensitivity refers to the proportion of people with a disease who are correctly identified as having the disease by the test. A test with high sensitivity will have few false negatives, meaning that it will rarely miss cases of the disease.

**Usage**

```
calc_tpr(...)

## Default S3 method:
calc_tpr(tp, fn, ci.type, ci.level, ...)

## S3 method for class 'table'
```

```
calc_tpr(tbl, ci.type, ci.level, ...)
```

```
## S3 method for class 'data.frame'
```

```
calc_tpr(data, prediction, reference, ci.type, ci.level, ...)
```

### Arguments

...	Additional arguments. Not used.
tp	Numeric, True Positives (TP).
fn	Numeric, False Negatives (FN).
ci.type	Either FALSE if no confidence intervals are desired or one of "agresti.coull", "agresti-coull", "ac", "asymptotic", "normal", "wald", "clopper-pearson", "cp", "exact", "jeffreys", "bayes", and "wilson". If FALSE, overwrites ci.level.
ci.level	A number between 0 and 1 for the levels of the confidence intervals that should be calculated.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

### Details

For example, a test for a certain type of cancer may have a sensitivity of 95%, which means that out of 100 people with the cancer, the test will correctly identify 95 of them as having the disease. This means that the test will have a 5% false negative rate, which means that out of 100 people with the cancer, the test will fail to detect the disease in 5 of them.

### Methods (by class)

- calc\_tpr(default):
- calc\_tpr(table):
- calc\_tpr(data.frame):

---

calc_tpr_macro	<i>Calculate TPR Macro.</i>
----------------	-----------------------------

---

### Description

Calculate TPR Macro.

**Usage**

```
calc_tpr_macro(...)

## Default S3 method:
calc_tpr_macro(tp, fn, ...)

## S3 method for class 'table'
calc_tpr_macro(tbl, ...)

## S3 method for class 'data.frame'
calc_tpr_macro(data, prediction, reference, ...)
```

**Arguments**

...	Additional arguments. Not used.
tp	Numeric vector of True Positives (TP) by class.
fn	Numeric vector of False Negatives (FN) by class.
tbl	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in rmetrics will generate incorrect results.
data	A data.frame containing the prediction and the reference.
prediction	Character. The name of the variable in data that contains the predictions.
reference	Character. The name of the variable in data that contains the reference values.

**Methods (by class)**

- calc\_tpr\_macro(default):
- calc\_tpr\_macro(table):
- calc\_tpr\_macro(data.frame):

---

calc_zero_one_loss	<i>Calculate Zero-One Loss.</i>
--------------------	---------------------------------

---

**Description**

Calculate Zero-One Loss.

**Usage**

```
calc_zero_one_loss(...)

## Default S3 method:
calc_zero_one_loss(otp, n, ...)

## S3 method for class 'table'
calc_zero_one_loss(tbl, ...)

## S3 method for class 'data.frame'
calc_zero_one_loss(data, prediction, reference, ...)
```

**Arguments**

<code>...</code>	Additional arguments. Not used.
<code>otp</code>	Overall True Positives (OTP).
<code>n</code>	Total number of observations.
<code>tbl</code>	A table representing the input confusion matrix. This must always have prediction on rows and reference on columns, otherwise most functions in <code>rmetrics</code> will generate incorrect results.
<code>data</code>	A <code>data.frame</code> containing the prediction and the reference.
<code>prediction</code>	Character. The name of the variable in <code>data</code> that contains the predictions.
<code>reference</code>	Character. The name of the variable in <code>data</code> that contains the reference values.

**Methods (by class)**

- `calc_zero_one_loss(default):`
- `calc_zero_one_loss(table):`
- `calc_zero_one_loss(data.frame):`

# Index

`calc_acc`, 3  
`calc_acc_macro`, 4  
`calc_af`, 5  
`calc_aickin`, 6  
`calc_aunp`, 7  
`calc_aunu`, 8  
`calc_aupr`, 9  
`calc_auroc`, 10  
`calc_b`, 11  
`calc_bbs`, 12  
`calc_bcd`, 13  
`calc_bennett_s`, 15  
`calc_cba`, 15  
`calc_chisq`, 16  
`calc_conditional_entropy`, 17  
`calc_cramer_v`, 18  
`calc_cross_entropy`, 19  
`calc_csi_macro`, 20  
`calc_dind`, 21  
`calc_dor`, 22  
`calc_dp`, 23  
`calc_err`, 24  
`calc_f`, 25  
`calc_f_macro`, 35  
`calc_fdr`, 26  
`calc_fmi`, 27  
`calc_fnr`, 28  
`calc_fnr_macro`, 29  
`calc_fnr_micro`, 30  
`calc_for`, 31  
`calc_fpr`, 32  
`calc_fpr_macro`, 33  
`calc_fpr_micro`, 34  
`calc_gini`, 36  
`calc_gmean`, 37  
`calc_grey`, 38  
`calc_gwet_ac1`, 39  
`calc_hamming`, 40  
`calc_iba`, 41  
`calc_icsi`, 42  
`calc_informedness`, 43  
`calc_is`, 44  
`calc_jaccard`, 45  
`calc_jaccard_overall`, 46  
`calc_joint_entropy`, 47  
`calc_kalpha`, 47  
`calc_kappa`, 48  
`calc_kl_divergence`, 49  
`calc_lambda`, 50  
`calc_lift`, 52  
`calc_markedness`, 53  
`calc_mcc`, 54  
`calc_mcc_overall`, 55  
`calc_mutual_information`, 56  
`calc_net_benefit`, 56  
`calc_nir`, 57  
`calc_nlr`, 58  
`calc_npv`, 59  
`calc_oacc`, 60  
`calc_oc`, 61  
`calc_ooc`, 62  
`calc_op`, 63  
`calc_oracc`, 64  
`calc_pearson_c`, 65  
`calc_phi`, 65  
`calc_phisq`, 66  
`calc_plr`, 67  
`calc_ppv_macro`, 68  
`calc_precision`, 69  
`calc_prevalence`, 70  
`calc_q`, 71  
`calc_racc`, 72  
`calc_rand`, 73  
`calc_rci`, 74  
`calc_reference_entropy`, 75  
`calc_response_entropy`, 76  
`calc_rr`, 77  
`calc_sind`, 78  
`calc_tnr`, 79  
`calc_tnr_macro`, 80  
`calc_tnr_micro`, 80  
`calc_tpr`, 81  
`calc_tpr_macro`, 82  
`calc_zero_one_loss`, 83