

Package ‘mlr3measures’

June 27, 2020

Title Performance Measures for ‘mlr3’

Version 0.2.0

Description Implements multiple performance measures for supervised learning. Includes over 40 measures for regression and classification. Additionally, meta information about the performance measures can be queried, e.g. what the best and worst possible performances scores are.

License LGPL-3

URL <https://mlr3measures.mlr-org.com>,
<https://github.com/mlr-org/mlr3measures>

BugReports <https://github.com/mlr-org/mlr3measures/issues>

Depends R (>= 3.1.0)

Imports checkmate

Suggests bibtex, testthat

Encoding UTF-8

RoxygenNote 7.1.0

Collate 'assertions.R' 'measures.R' 'binary_auc.R' 'binary_bbrier.R'
'binary_dor.R' 'binary_fbeta.R' 'binary_fdr.R' 'binary_fn.R'
'binary_fnr.R' 'binary_fomr.R' 'binary_fp.R' 'binary_fpr.R'
'binary_mcc.R' 'binary_npv.R' 'binary_ppv.R' 'binary_tn.R'
'binary_tnr.R' 'binary_tp.R' 'binary_tpr.R' 'classif_acc.R'
'classif_auc.R' 'classif_bacc.R' 'classif_ce.R'
'classif_logloss.R' 'classif_mbrier.R' 'confusion_matrix.R'
'helper.R' 'regr_bias.R' 'regr_ktau.R' 'regr_mae.R'
'regr_mape.R' 'regr_maxae.R' 'regr_maxse.R' 'regr_medae.R'
'regr_medse.R' 'regr_mse.R' 'regr_msle.R' 'regr_pbias.R'
'regr_rae.R' 'regr_rmse.R' 'regr_rmsle.R' 'regr_rrse.R'
'regr_rse.R' 'regr_rsq.R' 'regr_sae.R' 'regr_smape.R'
'regr_srho.R' 'regr_sse.R' 'roxygen.R' 'zzz.R'

NeedsCompilation no

Author Michel Lang [cre, aut] (<<https://orcid.org/0000-0001-9754-0393>>),
Martin Binder [ctb]

Maintainer Michel Lang <michellang@gmail.com>

Repository CRAN

Date/Publication 2020-06-27 04:50:02 UTC

R topics documented:

mlr3measures-package	3
acc	3
auc	4
bacc	6
bbrier	7
bias	8
ce	9
confusion_matrix	10
dor	11
fbeta	13
fdr	14
fn	16
fnr	17
fomr	18
fp	19
fpr	21
ktau	22
logloss	23
mae	24
mape	25
mauc_aunu	26
maxae	28
maxse	29
mbrier	30
mcc	31
measures	33
medae	33
medse	34
mse	35
msle	36
npv	37
pbias	39
ppv	40
rae	41
rmse	43
rmsle	44
rrse	45
rse	46
rsq	47
sae	48
smape	49

srho	51
sse	52
tn	53
tnr	54
tp	55
tpr	56
Index	59

mlr3measures-package *mlr3measures: Performance Measures for 'mlr3'*

Description

Implements multiple performance measures for supervised learning. Includes over 40 measures for regression and classification. Additionally, meta information about the performance measures can be queried, e.g. what the best and worst possible performances scores are.

Author(s)

Maintainer: Michel Lang <michellang@gmail.com> ([ORCID](#))

Other contributors:

- Martin Binder <mlr.developer@mb706.com> [contributor]

See Also

Useful links:

- <https://mlr3measures.ml-org.com>
- <https://github.com/mlr-org/mlr3measures>
- Report bugs at <https://github.com/mlr-org/mlr3measures/issues>

acc *Classification Accuracy*

Description

Classification measure defined as

$$\frac{1}{n} \sum_{i=1}^n (t_i = r_i).$$

Usage

`acc(truth, response, ...)`

Arguments

`truth` :: `factor()`
True (observed) labels. Must have the same levels and length as response.

`response` :: `factor()`
Predicted response labels. Must have the same levels and length as truth.

`...` :: `any`
Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "classif"
- Range: [0, 1]
- Minimize: FALSE
- Required prediction: response

See Also

Other Classification Measures: [bacc\(\)](#), [ce\(\)](#), [logloss\(\)](#), [mauc_aunu\(\)](#), [mbrier\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b", "c")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
acc(truth, response)
```

auc

Area Under the ROC Curve

Description

Computes the area under the Receiver Operator Characteristic (ROC) curve. The AUC can be interpreted as the probability that a randomly chosen positive observation has a higher predicted probability than a randomly chosen negative observation.

Usage

```
auc(truth, prob, positive, na_value = NaN, ...)
```

Arguments

<code>truth</code>	:: <code>factor()</code> True (observed) labels. Must have the exactly same two levels and the same length as response.
<code>prob</code>	:: <code>numeric()</code> Predicted probability for positive class. Must have exactly same length as <code>truth</code> .
<code>positive</code>	:: <code>character(1)</code> Name of the positive class.
<code>na_value</code>	:: <code>numeric(1)</code> Value that should be returned if the measure is not defined for the input (as described in the note). Default is <code>NaN</code> .
<code>...</code>	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: [0, 1]
- Minimize: FALSE
- Required prediction: `prob`

Note

This measure is undefined if the true values are either all positive or all negative.

References

Youden WJ (1950). "Index for rating diagnostic tests." *Cancer*, 3(1), 32–35. doi: [10.1002/1097-0142\(1950\)3:1<32::aid-cnrcr2820030106>3.0.co;23](https://doi.org/10.1002/1097-0142(1950)3:1<32::aid-cnrcr2820030106>3.0.co;23).

See Also

Other Binary Classification Measures: [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
truth = factor(c("a", "a", "a", "b"))
prob = c(.6, .7, .1, .4)
auc(truth, prob, "a")
```

bacc

*Balanced Accuracy***Description**

Computes the weighted balanced accuracy, suitable for imbalanced data sets. It is defined analogously to the definition in [sklearn](#).

First, the sample weights w are normalized per class:

$$\hat{w}_i = \frac{w_i}{\sum_j 1(y_j = y_i)w_i}.$$

The balanced accuracy is calculated as

$$\frac{1}{\sum_i \hat{w}_i} \sum_i 1(r_i = t_i)\hat{w}_i.$$

Usage

```
bacc(truth, response, sample_weights = NULL, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the same levels and length as response.
response	:: factor() Predicted response labels. Must have the same levels and length as truth.
sample_weights	:: numeric() Non-negative sample weights. Must have the same levels and length as truth. Defaults to equal sample weights.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "classif"
- Range: [0, 1]
- Minimize: FALSE
- Required prediction: response

References

Brodersen KH, Ong CS, Stephan KE, Buhmann JM (2010). “The Balanced Accuracy and Its Posterior Distribution.” In *2010 20th International Conference on Pattern Recognition*. doi: [10.1109/icpr.2010.764](https://doi.org/10.1109/icpr.2010.764).

Guyon I, Bennett K, Cawley G, Escalante HJ, Escalera S, Ho TK, Macia N, Ray B, Saeed M, Statnikov A, Viegas E (2015). “Design of the 2015 ChaLearn AutoML challenge.” In *2015 International Joint Conference on Neural Networks (IJCNN)*. doi: [10.1109/ijcnn.2015.7280767](https://doi.org/10.1109/ijcnn.2015.7280767).

See Also

Other Classification Measures: [acc\(\)](#), [ce\(\)](#), [logloss\(\)](#), [mauc_aunu\(\)](#), [mbrier\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b", "c")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
bacc(truth, response)
```

 bbrier

Binary Brier Score

Description

Brier score for binary classification problems defined as

$$\frac{1}{n} \sum_{i=1}^n (I_i - p_i)^2.$$

I_i is 1 if observation i belongs to the positive class, and 0 otherwise.

Note that this (more common) definition of the Brier score is equivalent to the original definition of the multi-class Brier score (see [mbrier\(\)](#)) divided by 2.

Usage

```
bbrier(truth, prob, positive, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the exactly same two levels and the same length as response.
prob	:: numeric() Predicted probability for positive class. Must have exactly same length as truth.
positive	:: character(1) Name of the positive class.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: [0, 1]
- Minimize: TRUE
- Required prediction: prob

References

https://en.wikipedia.org/wiki/Brier_score

Brier GW (1950). "Verification of forecasts expressed in terms of probability." *Monthly Weather Review*, **78**(1), 1–3. doi: [10.1175/15200493\(1950\)078<0001:vofeit>2.0.co;2](https://doi.org/10.1175/15200493(1950)078<0001:vofeit>2.0.co;2).

See Also

Other Binary Classification Measures: [auc\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
prob = runif(10)
bbrier(truth, prob, positive = "a")
```

bias

Bias

Description

Regression measure defined as

$$\frac{1}{n} \sum_{i=1}^n (t_i - r_i).$$

Good predictions score close to 0.

Usage

`bias(truth, response, ...)`

Arguments

truth :: numeric()
 True (observed) values. Must have the same length as response.

response :: numeric()
 Predicted response values. Must have the same length as truth.

... :: any
 Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $(-\infty, \infty)$
- Minimize: NA
- Required prediction: response

See Also

Other Regression Measures: [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
bias(truth, response)
```

ce

Classification Error

Description

Classification measure defined as

$$\frac{1}{n} \sum_{i=1}^n (t_i \neq r_i).$$

Usage

```
ce(truth, response, ...)
```

Arguments

truth :: factor()
True (observed) labels. Must have the same levels and length as response.

response :: factor()
Predicted response labels. Must have the same levels and length as truth.

... :: any
Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "classif"
- Range: [0, 1]
- Minimize: TRUE
- Required prediction: response

See Also

Other Classification Measures: [acc\(\)](#), [bacc\(\)](#), [logloss\(\)](#), [mauc_aunu\(\)](#), [mbrier\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b", "c")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
ce(truth, response)
```

confusion_matrix

Calculate Binary Confusion Matrix

Description

Calculates the confusion matrix for a binary classification problem once and then calculates all confusion measures of this package.

Usage

```
confusion_matrix(truth, response, positive, na_value = NaN, relative = FALSE)
```

Arguments

truth	:: factor() True (observed) labels. Must have the exactly same two levels and the same length as response.
response	:: factor() Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	:: character(1) Name of the positive class.
na_value	:: numeric(1) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
relative	:: logical(1) If TRUE, the returned confusion matrix contains relative frequencies instead of absolute frequencies.

Value

List with two elements:

- `matrix` stores the calculated confusion matrix.
- `measures` stores the metrics as named numeric vector.

Examples

```
set.seed(123)
lvls = c("a", "b")
truth = factor(sample(lvls, 20, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 20, replace = TRUE), levels = lvls)

confusion_matrix(truth, response, positive = "a")
confusion_matrix(truth, response, positive = "a", relative = TRUE)
confusion_matrix(truth, response, positive = "b")
```

dor

Diagnostic Odds Ratio

Description

Binary classification measure defined as

$$\frac{TP/FP}{FN/TN}$$

Usage

```
dor(truth, response, positive, na_value = NaN, ...)
```

Arguments

<code>truth</code>	:: <code>factor()</code> True (observed) labels. Must have the exactly same two levels and the same length as response.
<code>response</code>	:: <code>factor()</code> Predicted response labels. Must have the exactly same two levels and the same length as <code>truth</code> .
<code>positive</code>	:: <code>character(1)</code> Name of the positive class.
<code>na_value</code>	:: <code>numeric(1)</code> Value that should be returned if the measure is not defined for the input (as described in the note). Default is <code>NaN</code> .
<code>...</code>	:: <code>any</code> Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: $[0, \infty)$
- Minimize: FALSE
- Required prediction: response

Note

This measure is undefined if $FP = 0$ or $FN = 0$.

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
dor(truth, response, positive = "a")
```

fbeta	<i>F-beta Score</i>
-------	---------------------

Description

Binary classification measure defined with P as `precision()` and R as `recall()` as

$$(1 + \beta^2) \frac{P \cdot R}{(\beta^2 P) + R}.$$

It measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision. For $\beta = 1$, this measure is called "F1" score.

Usage

```
fbeta(truth, response, positive, beta = 1, na_value = NaN, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the exactly same two levels and the same length as response.
response	:: factor() Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	:: character(1) Name of the positive class.
beta	:: numeric(1) Parameter to give either precision or recall more weight. Default is 1, resulting in balanced weights.
na_value	:: numeric(1) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: [0, 1]
- Minimize: FALSE
- Required prediction: response

Note

This measure is undefined if

- TP = 0
- [precision](#) or [recall](#) is undefined, i.e. TP + FP = 0 or TP + FN = 0.

References

Sasaki Y, others (2007). "The truth of the F-measure." *Teach Tutor mater*, **1**(5), 1–5. <https://www.cs.odu.edu/~mukka/cs795sum10dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf>.

Rijsbergen CJV (1979). *Information Retrieval*, 2nd edition. Butterworth-Heinemann, Newton, MA, USA. ISBN 408709294.

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fbeta(truth, response, positive = "a")
```

fdr

False Discovery Rate

Description

Binary classification measure defined as

$$\frac{FP}{TP + FP}$$

Usage

```
fdr(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the exactly same two levels and the same length as response.
response	:: factor() Predicted response labels. Must have the exactly same two levels and the same length as truth.

```
positive      :: character(1)
               Name of the positive class.

na_value      :: numeric(1)
               Value that should be returned if the measure is not defined for the input (as
               described in the note). Default is NaN.

...           :: any
               Additional arguments. Currently ignored.
```

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: [0, 1]
- Minimize: TRUE
- Required prediction: response

Note

This measure is undefined if $TP + FP = 0$.

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: `auc()`, `bbrier()`, `dor()`, `fbeta()`, `fnr()`, `fn()`, `fomr()`, `fpr()`, `fp()`, `mcc()`, `npv()`, `ppv()`, `tnr()`, `tn()`, `tpr()`, `tp()`

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fdr(truth, response, positive = "a")
```

fn	<i>False Negatives</i>
----	------------------------

Description

Classification measure counting the false negatives (type 2 error), i.e. the number of predictions indicating a negative class label while in fact it is positive. This is sometimes also called a "false alarm".

Usage

```
fn(truth, response, positive, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the exactly same two levels and the same length as response.
response	:: factor() Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	:: character(1) Name of the positive class.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fn(truth, response, positive = "a")
```

fnr	<i>False Negative Rate</i>
-----	----------------------------

Description

Binary classification measure defined as

$$\frac{FN}{TP + FN}$$

Also know as "miss rate".

Usage

```
fnr(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	:: factor()	True (observed) labels. Must have the exactly same two levels and the same length as response.
response	:: factor()	Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	:: character(1)	Name of the positive class.
na_value	:: numeric(1)	Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any	Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: [0, 1]
- Minimize: TRUE
- Required prediction: response

Note

This measure is undefined if $TP + FN = 0$.

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fmr(truth, response, positive = "a")
```

fomr

False Omission Rate

Description

Binary classification measure defined as

$$\frac{FN}{FN + TN}$$

Usage

```
fomr(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the exactly same two levels and the same length as response.
response	:: factor() Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	:: character(1) Name of the positive class.
na_value	:: numeric(1) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: [0, 1]
- Minimize: TRUE
- Required prediction: response

Note

This measure is undefined if $FN + TN = 0$.

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fomr(truth, response, positive = "a")
```

fp

False Positives

Description

Classification measure counting the false positives (type 1 error), i.e. the number of predictions indicating a positive class label while in fact it is negative.

Usage

```
fp(truth, response, positive, ...)
```

Arguments

<code>truth</code>	:: <code>factor()</code> True (observed) labels. Must have the exactly same two levels and the same length as response.
<code>response</code>	:: <code>factor()</code> Predicted response labels. Must have the exactly same two levels and the same length as truth.
<code>positive</code>	:: <code>character(1)</code> Name of the positive class.
<code>...</code>	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fp(truth, response, positive = "a")
```

fpr *False Positive Rate*

Description

Binary classification measure defined as

$$\frac{FP}{FP + TN}$$

Also know as fall out or probability of false alarm.

Usage

```
fpr(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the exactly same two levels and the same length as response.
response	:: factor() Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	:: character(1) Name of the positive class.
na_value	:: numeric(1) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as numeric(1).

Meta Information

- Type: "binary"
- Range: [0, 1]
- Minimize: TRUE
- Required prediction: response

Note

This measure is undefined if $FP + TN = 0$.

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
fpr(truth, response, positive = "a")
```

ktau

Kendall's tau

Description

Regression measure defined as Kendall's rank correlation coefficient between truth and response. Calls `stats::cor()` with method set to "kendall".

Usage

```
ktau(truth, response, ...)
```

Arguments

truth	:: numeric()	True (observed) values. Must have the same length as response.
response	:: numeric()	Predicted response values. Must have the same length as truth.
...	:: any	Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[-1, 1]$
- Minimize: FALSE
- Required prediction: response

References

Rosset S, Perlich C, Zadrozny B (2006). "Ranking-based evaluation of regression models." *Knowledge and Information Systems*, **12**(3), 331–353. doi: [10.1007/s1011500600373](https://doi.org/10.1007/s1011500600373).

See Also

Other Regression Measures: [bias\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
ktau(truth, response)
```

logloss	<i>Log Loss</i>
---------	-----------------

Description

Classification measure defined as

$$-\frac{1}{n} \sum_{i=1}^n \log(p_i)$$

where p_i is the probability for the true class of observation i .

Usage

```
logloss(truth, prob, eps = 1e-15, ...)
```

Arguments

truth	:: factor()	True (observed) labels. Must have the same levels and length as response.
prob	:: matrix()	Matrix of predicted probabilities, each column is a vector of probabilities for a specific class label. Columns must be named with levels of truth.
eps	:: numeric(1)	Probabilities are clipped to $\max(\text{eps}, \min(1 - \text{eps}, p))$. Otherwise the measure would be undefined for probabilities $p = 0$ and $p = 1$.
...	:: any	Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "classif"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: prob

See Also

Other Classification Measures: [acc\(\)](#), [bacc\(\)](#), [ce\(\)](#), [mauc_aunu\(\)](#), [mbrier\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b", "c")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
prob = matrix(runif(3 * 10), ncol = 3, dimnames = list(NULL, lvls))
prob = t(apply(prob, 1, function(x) x / sum(x)))
logloss(truth, prob)
```

mae

*Mean Absolute Errors***Description**

Regression measure defined as

$$\frac{1}{n} \sum_{i=1}^n |t_i - r_i|.$$

Usage

```
mae(truth, response, ...)
```

Arguments

truth	:: numeric() True (observed) values. Must have the same length as response.
response	:: numeric() Predicted response values. Must have the same length as truth.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
mae(truth, response)
```

mape

Mean Absolute Percent Error

Description

Regression measure defined as

$$\frac{1}{n} \sum_{i=1}^n \left| \frac{t_i - r_i}{t_i} \right|.$$

Usage

```
mape(truth, response, na_value = NaN, ...)
```

Arguments

truth	:: numeric()	True (observed) values. Must have the same length as response.
response	:: numeric()	Predicted response values. Must have the same length as truth.
na_value	:: numeric(1)	Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any	Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

Note

This measure is undefined if any element of t is 0.

References

de Myttenaere A, Golden B, Le Grand B, Rossi F (2016). "Mean Absolute Percentage Error for regression models." *Neurocomputing*, **192**, 38–48. ISSN 0925-2312, doi: [10.1016/j.neucom.2015.12.114](https://doi.org/10.1016/j.neucom.2015.12.114).

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
mape(truth, response)
```

mauc_aunu

Multiclass AUC Scores

Description

Multiclass AUC measures.

- *AUNU*: AUC of each class against the rest, using the uniform class distribution. Computes the AUC treating a c -dimensional classifier as c two-dimensional 1-vs-rest classifiers, where classes are assumed to have uniform distribution, in order to have a measure which is independent of class distribution change (Fawcett 2001).
- *AUNP*: AUC of each class against the rest, using the a priori class distribution. Computes the AUC treating a c -dimensional classifier as c two-dimensional 1-vs-rest classifiers, taking into account the prior probability of each class (Fawcett 2001).

- *AUIU*: AUC of each class against each other, using the uniform class distribution. Computes something like the AUC of $c(c-1)$ binary classifiers (all possible pairwise combinations). See Hand (2001) for details.
- *AUIP*: AUC of each class against each other, using the a priori class distribution. Computes something like AUC of $c(c-1)$ binary classifiers while considering the a priori distribution of the classes as suggested in Ferri (2009). Note we deviate from the definition in Ferri (2009) by a factor of c . The person implementing this function and writing this very documentation right now cautions against using this measure because it is an imperfect generalization of AUIU.

Usage

```
mauc_aunu(truth, prob, na_value = NaN, ...)
```

```
mauc_aunp(truth, prob, na_value = NaN, ...)
```

```
mauc_auiu(truth, prob, na_value = NaN, ...)
```

```
mauc_auip(truth, prob, na_value = NaN, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the same levels and length as response.
prob	:: matrix() Matrix of predicted probabilities, each column is a vector of probabilities for a specific class label. Columns must be named with levels of truth.
na_value	:: numeric(1) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "classif"
- Range: [0, 1]
- Minimize: FALSE
- Required prediction: prob

References

Fawcett T (2001). "Using rule sets to maximize ROC performance." In *Proceedings 2001 IEEE international conference on data mining*, 131–138. IEEE.

Ferri C, Hernández-Orallo J, Modroiu R (2009). “An experimental comparison of performance measures for classification.” *Pattern Recognition Letters*, **30**(1), 27–38. doi: [10.1016/j.patrec.2008.08.010](https://doi.org/10.1016/j.patrec.2008.08.010).

Hand DJ, Till RJ (2001). “A simple generalisation of the area under the ROC curve for multiple class classification problems.” *Machine learning*, **45**(2), 171–186.

See Also

Other Classification Measures: `acc()`, `bacc()`, `ce()`, `logloss()`, `mbrier()`

Examples

```
set.seed(1)
lvls = c("a", "b", "c")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
prob = matrix(runif(3 * 10), ncol = 3)
colnames(prob) = levels(truth)
mauc_aunu(truth, prob)
```

maxae

Max Absolute Error

Description

Regression measure defined as

$$\max(|t_i - r_i|).$$

Usage

```
maxae(truth, response, ...)
```

Arguments

<code>truth</code>	:: numeric() True (observed) values. Must have the same length as response.
<code>response</code>	:: numeric() Predicted response values. Must have the same length as truth.
<code>...</code>	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
maxae(truth, response)
```

maxse

Max Squared Error

Description

Regression measure defined as

$$\max (t_i - r_i)^2 .$$

Usage

```
maxse(truth, response, ...)
```

Arguments

truth	:: numeric()	True (observed) values. Must have the same length as response.
response	:: numeric()	Predicted response values. Must have the same length as truth.
...	:: any	Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
maxse(truth, response)
```

mbrier

Multiclass Brier Score

Description

Brier score for multi-class classification problems with r labels defined as

$$\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^r (I_{ij} - p_{ij})^2.$$

I_{ij} is 1 if observation i has true label j , and 0 otherwise.

Note that there also is the more common definition of the Brier score for binary classification problems in [bbrier\(\)](#).

Usage

```
mbrier(truth, prob, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the same levels and length as response.
prob	:: matrix() Matrix of predicted probabilities, each column is a vector of probabilities for a specific class label. Columns must be named with levels of truth.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "classif"
- Range: [0, 2]
- Minimize: TRUE
- Required prediction: prob

References

Brier GW (1950). "Verification of forecasts expressed in terms of probability." *Monthly Weather Review*, **78**(1), 1–3. doi: [10.1175/15200493\(1950\)078<0001:vofeit>2.0.co;2](https://doi.org/10.1175/15200493(1950)078<0001:vofeit>2.0.co;2).

See Also

Other Classification Measures: [acc\(\)](#), [bacc\(\)](#), [ce\(\)](#), [logloss\(\)](#), [mauc_aunu\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b", "c")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
prob = matrix(runif(3 * 10), ncol = 3)
colnames(prob) = levels(truth)
mbrier(truth, prob)
```

mcc

*Matthews Correlation Coefficient***Description**

Binary classification measure defined as

$$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Usage

```
mcc(truth, response, positive, ...)
```

Arguments

`truth` :: `factor()`
True (observed) labels. Must have the exactly same two levels and the same length as response.

`response` :: `factor()`
Predicted response labels. Must have the exactly same two levels and the same length as truth.

```
positive      :: character(1)
               Name of the positive class.

...          :: any
               Additional arguments. Currently ignored.
```

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: $[-1, 1]$
- Minimize: FALSE
- Required prediction: response

Note

This above formula is undefined if any of the four sums in the denominator is 0. The denominator is then set to 1.

References

Matthews BW (1975). "Comparison of the predicted and observed secondary structure of T4 phage lysozyme." *Biochimica et Biophysica Acta (BBA) - Protein Structure*, **405**(2), 442–451. doi: [10.1016/00052795\(75\)901099](https://doi.org/10.1016/00052795(75)901099).

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
mcc(truth, response, positive = "a")
```

measures *Measure Registry*

Description

The `environment()` measures keeps track of all measures in this package. Stores meta information about measures, such as minimum, maximum of if the measure must be minimized or maximized.

Usage

```
measures
```

Format

An object of class `environment` of length 51.

Examples

```
names(measures)
measures$tp
```

medae *Median Absolute Errors*

Description

Regression measure defined as

$$\operatorname{median}_i |t_i - r_i|.$$

Usage

```
medae(truth, response, ...)
```

Arguments

<code>truth</code>	:: <code>numeric()</code> True (observed) values. Must have the same length as <code>response</code> .
<code>response</code>	:: <code>numeric()</code> Predicted response values. Must have the same length as <code>truth</code> .
<code>...</code>	:: <code>any</code> Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
medae(truth, response)
```

medse

Median Squared Error

Description

Regression measure defined as

$$\text{median}_i \left[(t_i - r_i)^2 \right].$$

Usage

```
medse(truth, response, ...)
```

Arguments

truth	:: numeric() True (observed) values. Must have the same length as response.
response	:: numeric() Predicted response values. Must have the same length as truth.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
medse(truth, response)
```

mse

Mean Squared Error

Description

Regression measure defined as

$$\frac{1}{n} \sum_{i=1}^n (t_i - r_i)^2.$$

Usage

```
mse(truth, response, ...)
```

Arguments

truth	:: numeric() True (observed) values. Must have the same length as response.
response	:: numeric() Predicted response values. Must have the same length as truth.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
mse(truth, response)
```

msle

Mean Squared Log Error

Description

Regression measure defined as

$$\frac{1}{n} \sum_{i=1}^n (\ln(1 + t_i) - \ln(1 + r_i))^2.$$

Usage

```
msle(truth, response, na_value = NaN, ...)
```

Arguments

truth	:: numeric() True (observed) values. Must have the same length as response.
response	:: numeric() Predicted response values. Must have the same length as truth.
na_value	:: numeric(1) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

Note

This measure is undefined if any element of t or r is less than or equal to -1 .

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
msle(truth, response)
```

npv

Negative Predictive Value

Description

Binary classification measure defined as

$$\frac{TN}{FN + TN}$$

Usage

```
npv(truth, response, positive, na_value = NaN, ...)
```

Arguments

<code>truth</code>	:: <code>factor()</code> True (observed) labels. Must have the exactly same two levels and the same length as response.
<code>response</code>	:: <code>factor()</code> Predicted response labels. Must have the exactly same two levels and the same length as <code>truth</code> .
<code>positive</code>	:: <code>character(1)</code> Name of the positive class.
<code>na_value</code>	:: <code>numeric(1)</code> Value that should be returned if the measure is not defined for the input (as described in the note). Default is <code>NaN</code> .
<code>...</code>	:: <code>any</code> Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: [0, 1]
- Minimize: FALSE
- Required prediction: response

Note

This measure is undefined if $FN + TN = 0$.

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
npv(truth, response, positive = "a")
```

pbias

*Percent Bias***Description**

Regression measure defined as

$$\frac{1}{n} \sum_{i=1}^n \frac{(t_i - r_i)}{|t_i|}.$$

Good predictions score close to 0.

Usage

```
pbias(truth, response, na_value = NaN, ...)
```

Arguments

truth	:: numeric()	True (observed) values. Must have the same length as response.
response	:: numeric()	Predicted response values. Must have the same length as truth.
na_value	:: numeric(1)	Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any	Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $(-\infty, \infty)$
- Minimize: NA
- Required prediction: response

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
pbias(truth, response)
```

 ppv

Positive Predictive Value

Description

Binary classification measure defined as

$$\frac{TP}{TP + FP}$$

Also know as "precision".

Usage

```
ppv(truth, response, positive, na_value = NaN, ...)
```

```
precision(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the exactly same two levels and the same length as response.
response	:: factor() Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	:: character(1) Name of the positive class.
na_value	:: numeric(1) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as numeric(1).

Meta Information

- Type: "binary"
- Range: [0, 1]
- Minimize: FALSE
- Required prediction: response

Note

This measure is undefined if $TP + FP = 0$.

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
ppv(truth, response, positive = "a")
```

rae

Relative Absolute Error

Description

Regression measure defined as

$$\frac{\sum_{i=1}^n |t_i - r_i|}{\sum_{i=1}^n |t_i - \bar{t}|}$$

Can be interpreted as absolute error of the predictions relative to a naive model predicting the mean.

Usage

```
rae(truth, response, na_value = NaN, ...)
```

Arguments

<code>truth</code>	:: <code>numeric()</code> True (observed) values. Must have the same length as <code>response</code> .
<code>response</code>	:: <code>numeric()</code> Predicted response values. Must have the same length as <code>truth</code> .
<code>na_value</code>	:: <code>numeric(1)</code> Value that should be returned if the measure is not defined for the input (as described in the note). Default is <code>NaN</code> .
<code>...</code>	:: <code>any</code> Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: `response`

Note

This measure is undefined for constant t .

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
rae(truth, response)
```

rmse *Root Mean Squared Error*

Description

Regression measure defined as

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (t_i - r_i)^2}.$$

Usage

```
rmse(truth, response, ...)
```

Arguments

truth	:: numeric()	True (observed) values. Must have the same length as response.
response	:: numeric()	Predicted response values. Must have the same length as truth.
...	:: any	Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
rmse(truth, response)
```

rmsle

*Root Mean Squared Log Error***Description**

Regression measure defined as

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (\ln(1 + t_i) - \ln(1 + r_i))^2}.$$

Usage

```
rmsle(truth, response, na_value = NaN, ...)
```

Arguments

truth	:: numeric()	True (observed) values. Must have the same length as response.
response	:: numeric()	Predicted response values. Must have the same length as truth.
na_value	:: numeric(1)	Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any	Additional arguments. Currently ignored.

Value

Performance value as numeric(1).

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

Note

This measure is undefined if any element of t or r is less than or equal to -1 .

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
rmsle(truth, response)
```

rrse

*Root Relative Squared Error***Description**

Regression measure defined as

$$\sqrt{\frac{\sum_{i=1}^n (t_i - r_i)^2}{\sum_{i=1}^n (t_i - \bar{t})^2}}$$

Can be interpreted as root of the squared error of the predictions relative to a naive model predicting the mean.

Usage

```
rrse(truth, response, na_value = NaN, ...)
```

Arguments

truth	:: numeric()	True (observed) values. Must have the same length as response.
response	:: numeric()	Predicted response values. Must have the same length as truth.
na_value	:: numeric(1)	Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any	Additional arguments. Currently ignored.

Value

Performance value as numeric(1).

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

Note

This measure is undefined for constant t .

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
rrse(truth, response)
```

rse

Relative Squared Error

Description

Regression measure defined as

$$\frac{\sum_{i=1}^n (t_i - r_i)^2}{\sum_{i=1}^n (t_i - \bar{t})^2}.$$

Can be interpreted as squared error of the predictions relative to a naive model predicting the mean.

Usage

```
rse(truth, response, na_value = NaN, ...)
```

Arguments

truth	:: numeric()	True (observed) values. Must have the same length as response.
response	:: numeric()	Predicted response values. Must have the same length as truth.
na_value	:: numeric(1)	Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any	Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

Note

This measure is undefined for constant t .

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
rse(truth, response)
```

rsq

R Squared

Description

Regression measure defined as

$$1 - \frac{\sum_{i=1}^n (t_i - r_i)^2}{\sum_{i=1}^n (t_i - \bar{t})^2}.$$

Also known as coefficient of determination or explained variation. Subtracts the [rse\(\)](#) from 1, hence it compares the squared error of the predictions relative to a naive model predicting the mean.

Usage

```
rsq(truth, response, na_value = NaN, ...)
```

Arguments

`truth` :: [numeric\(\)](#)
True (observed) values. Must have the same length as `response`.

`response` :: [numeric\(\)](#)
Predicted response values. Must have the same length as `truth`.

```
na_value      :: numeric(1)
                Value that should be returned if the measure is not defined for the input (as
                described in the note). Default is NaN.

...           :: any
                Additional arguments. Currently ignored.
```

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $(-\infty, 1]$
- Minimize: FALSE
- Required prediction: response

Note

This measure is undefined for constant t .

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [sae\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
rsq(truth, response)
```

sae

Sum of Absolute Errors

Description

Regression measure defined as

$$\sum_{i=1}^n |t_i - r_i|.$$

Usage

```
sae(truth, response, ...)
```


Arguments

truth :: numeric()
 True (observed) values. Must have the same length as response.

response :: numeric()
 Predicted response values. Must have the same length as truth.

... :: any
 Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [smape\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
sae(truth, response)
```

smape

Symmetric Mean Absolute Percent Error

Description

Regression measure defined as

$$\frac{2}{n} \sum_{i=1}^n \frac{|t_i - r_i|}{|t_i| + |r_i|}$$

Usage

```
smape(truth, response, na_value = NaN, ...)
```

Arguments

<code>truth</code>	:: <code>numeric()</code> True (observed) values. Must have the same length as <code>response</code> .
<code>response</code>	:: <code>numeric()</code> Predicted response values. Must have the same length as <code>truth</code> .
<code>na_value</code>	:: <code>numeric(1)</code> Value that should be returned if the measure is not defined for the input (as described in the note). Default is <code>NaN</code> .
<code>...</code>	:: <code>any</code> Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: [0, 2]
- Minimize: TRUE
- Required prediction: response

Note

This measure is undefined if if any $|t| + |r|$ is 0.

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [srho\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
smape(truth, response)
```

srho	<i>Spearman's rho</i>
------	-----------------------

Description

Regression measures defined as Spearman's rank correlation coefficient between truth and response. Calls `stats::cor()` with method set to "spearman".

Usage

```
srho(truth, response, ...)
```

Arguments

truth	:: numeric() True (observed) values. Must have the same length as response.
response	:: numeric() Predicted response values. Must have the same length as truth.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "regr"
- Range: $[-1, 1]$
- Minimize: FALSE
- Required prediction: response

References

Rosset S, Perlich C, Zadrozny B (2006). "Ranking-based evaluation of regression models." *Knowledge and Information Systems*, 12(3), 331–353. doi: [10.1007/s1011500600373](https://doi.org/10.1007/s1011500600373).

See Also

Other Regression Measures: [bias\(\)](#), [ktau\(\)](#), [mae\(\)](#), [mape\(\)](#), [maxae\(\)](#), [maxse\(\)](#), [medae\(\)](#), [medse\(\)](#), [mse\(\)](#), [msle\(\)](#), [pbias\(\)](#), [rae\(\)](#), [rmse\(\)](#), [rmsle\(\)](#), [rrse\(\)](#), [rse\(\)](#), [rsq\(\)](#), [sae\(\)](#), [smape\(\)](#), [sse\(\)](#)

Examples

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
srho(truth, response)
```

`sse`*Sum of Squared Errors*

Description

Regression measure defined as

$$\sum_{i=1}^n (t_i - r_i)^2.$$

Usage`sse(truth, response, ...)`**Arguments**

<code>truth</code>	:: <code>numeric()</code> True (observed) values. Must have the same length as <code>response</code> .
<code>response</code>	:: <code>numeric()</code> Predicted response values. Must have the same length as <code>truth</code> .
<code>...</code>	:: <code>any</code> Additional arguments. Currently ignored.

ValuePerformance value as `numeric(1)`.**Meta Information**

- Type: "regr"
- Range: $[0, \infty)$
- Minimize: TRUE
- Required prediction: response

See AlsoOther Regression Measures: `bias()`, `ktau()`, `mae()`, `mape()`, `maxae()`, `maxse()`, `medae()`, `medse()`, `mse()`, `msle()`, `pbias()`, `rae()`, `rmse()`, `rmsle()`, `rrse()`, `rse()`, `rsq()`, `sae()`, `smape()`, `srho()`**Examples**

```
set.seed(1)
truth = 1:10
response = truth + rnorm(10)
sse(truth, response)
```

tn	<i>True Negatives</i>
----	-----------------------

Description

Classification measure counting the true negatives, i.e. the number of predictions correctly indicating a negative class label.

Usage

```
tn(truth, response, positive, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the exactly same two levels and the same length as response.
response	:: factor() Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	:: character(1) Name of the positive class.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: $[0, \infty)$
- Minimize: FALSE
- Required prediction: response

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
tn(truth, response, positive = "a")
```

tnr

*True Negative Rate***Description**

Binary classification measure defined as

$$\frac{TN}{FP + TN}$$

Also know as "specificity".

Usage

```
tnr(truth, response, positive, na_value = NaN, ...)
specificity(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the exactly same two levels and the same length as response.
response	:: factor() Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	:: character(1) Name of the positive class.
na_value	:: numeric(1) Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any Additional arguments. Currently ignored.

Value

Performance value as numeric(1).

Meta Information

- Type: "binary"
- Range: [0, 1]
- Minimize: FALSE
- Required prediction: response

Note

This measure is undefined if $FP + TN = 0$.

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tn\(\)](#), [tpr\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
tnr(truth, response, positive = "a")
```

tp	<i>True Positives</i>
----	-----------------------

Description

Binary classification measure counting the true positives, i.e. the number of predictions correctly indicating a positive class label.

Usage

```
tp(truth, response, positive, ...)
```

Arguments

truth	:: factor() True (observed) labels. Must have the exactly same two levels and the same length as response.
response	:: factor() Predicted response labels. Must have the exactly same two levels and the same length as truth.

```

positive      :: character(1)
               Name of the positive class.

...          :: any
               Additional arguments. Currently ignored.

```

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: $[0, \infty)$
- Minimize: FALSE
- Required prediction: response

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tpr\(\)](#)

Examples

```

set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
tp(truth, response, positive = "a")

```

tpr

True Positive Rate

Description

Binary classification measure defined as

$$\frac{TP}{TP + FN}$$

Also know as "recall" or "sensitivity".

Usage

```
tpr(truth, response, positive, na_value = NaN, ...)

recall(truth, response, positive, na_value = NaN, ...)

sensitivity(truth, response, positive, na_value = NaN, ...)
```

Arguments

truth	:: factor()	True (observed) labels. Must have the exactly same two levels and the same length as response.
response	:: factor()	Predicted response labels. Must have the exactly same two levels and the same length as truth.
positive	:: character(1)	Name of the positive class.
na_value	:: numeric(1)	Value that should be returned if the measure is not defined for the input (as described in the note). Default is NaN.
...	:: any	Additional arguments. Currently ignored.

Value

Performance value as `numeric(1)`.

Meta Information

- Type: "binary"
- Range: [0, 1]
- Minimize: FALSE
- Required prediction: response

Note

This measure is undefined if $TP + FN = 0$.

References

https://en.wikipedia.org/wiki/Template:DiagnosticTesting_Diagram

See Also

Other Binary Classification Measures: [auc\(\)](#), [bbrier\(\)](#), [dor\(\)](#), [fbeta\(\)](#), [fdr\(\)](#), [fnr\(\)](#), [fn\(\)](#), [fomr\(\)](#), [fpr\(\)](#), [fp\(\)](#), [mcc\(\)](#), [npv\(\)](#), [ppv\(\)](#), [tnr\(\)](#), [tn\(\)](#), [tp\(\)](#)

Examples

```
set.seed(1)
lvls = c("a", "b")
truth = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
response = factor(sample(lvls, 10, replace = TRUE), levels = lvls)
tpr(truth, response, positive = "a")
```

Index

*Topic **datasets**

- measures, 33
- acc, 3, 7, 10, 24, 28, 31
- auc, 4, 8, 12, 14–16, 18–20, 22, 32, 38, 41, 53, 55–57
- bacc, 4, 6, 10, 24, 28, 31
- bbrier, 5, 7, 12, 14–16, 18–20, 22, 32, 38, 41, 53, 55–57
- bbrier(), 30
- bias, 8, 23, 25, 26, 29, 30, 34–37, 39, 42–44, 46–52
- ce, 4, 7, 9, 24, 28, 31
- confusion_matrix, 10
- dor, 5, 8, 11, 14–16, 18–20, 22, 32, 38, 41, 53, 55–57
- environment(), 33
- fbeta, 5, 8, 12, 13, 15, 16, 18–20, 22, 32, 38, 41, 53, 55–57
- fdr, 5, 8, 12, 14, 14, 16, 18–20, 22, 32, 38, 41, 53, 55–57
- fn, 5, 8, 12, 14, 15, 16, 18–20, 22, 32, 38, 41, 53, 55–57
- fnr, 5, 8, 12, 14–16, 17, 19, 20, 22, 32, 38, 41, 53, 55–57
- fomr, 5, 8, 12, 14–16, 18, 18, 20, 22, 32, 38, 41, 53, 55–57
- fp, 5, 8, 12, 14–16, 18, 19, 19, 22, 32, 38, 41, 53, 55–57
- fpr, 5, 8, 12, 14–16, 18–20, 21, 32, 38, 41, 53, 55–57
- ktau, 9, 22, 25, 26, 29, 30, 34–37, 39, 42–44, 46–52
- logloss, 4, 7, 10, 23, 28, 31
- mae, 9, 23, 24, 26, 29, 30, 34–37, 39, 42–44, 46–52
- mape, 9, 23, 25, 25, 29, 30, 34–37, 39, 42–44, 46–52
- mauc_au1p (mauc_aunu), 26
- mauc_au1u (mauc_aunu), 26
- mauc_aunp (mauc_aunu), 26
- mauc_aunu, 4, 7, 10, 24, 26, 31
- maxae, 9, 23, 25, 26, 28, 30, 34–37, 39, 42–44, 46–52
- maxse, 9, 23, 25, 26, 29, 29, 34–37, 39, 42–44, 46–52
- mbrier, 4, 7, 10, 24, 28, 30
- mbrier(), 7
- mcc, 5, 8, 12, 14–16, 18–20, 22, 31, 38, 41, 53, 55–57
- measures, 33
- medae, 9, 23, 25, 26, 29, 30, 33, 35–37, 39, 42–44, 46–52
- medse, 9, 23, 25, 26, 29, 30, 34, 34, 36, 37, 39, 42–44, 46–52
- mlr3measures (mlr3measures-package), 3
- mlr3measures-package, 3
- mse, 9, 23, 25, 26, 29, 30, 34, 35, 35, 37, 39, 42–44, 46–52
- msle, 9, 23, 25, 26, 29, 30, 34–36, 36, 39, 42–44, 46–52
- npv, 5, 8, 12, 14–16, 18–20, 22, 32, 37, 41, 53, 55–57
- pbias, 9, 23, 25, 26, 29, 30, 34–37, 39, 42–44, 46–52
- ppv, 5, 8, 12, 14–16, 18–20, 22, 32, 38, 40, 53, 55–57
- precision, 14
- precision (ppv), 40
- precision(), 13
- rae, 9, 23, 25, 26, 29, 30, 34–37, 39, 41, 43, 44, 46–52

recall, [14](#)
recall (tpr), [56](#)
recall(), [13](#)
rmse, [9](#), [23](#), [25](#), [26](#), [29](#), [30](#), [34–37](#), [39](#), [42](#), [43](#),
[44](#), [46–52](#)
rmsle, [9](#), [23](#), [25](#), [26](#), [29](#), [30](#), [34–37](#), [39](#), [42](#), [43](#),
[44](#), [46–52](#)
rrse, [9](#), [23](#), [25](#), [26](#), [29](#), [30](#), [34–37](#), [39](#), [42–44](#),
[45](#), [47–52](#)
rse, [9](#), [23](#), [25](#), [26](#), [29](#), [30](#), [34–37](#), [39](#), [42–44](#),
[46](#), [46](#), [48–52](#)
rse(), [47](#)
rsq, [9](#), [23](#), [25](#), [26](#), [29](#), [30](#), [34–37](#), [39](#), [42–44](#),
[46](#), [47](#), [47](#), [49–52](#)

sae, [9](#), [23](#), [25](#), [26](#), [29](#), [30](#), [34–37](#), [39](#), [42–44](#),
[46–48](#), [48](#), [50–52](#)
sensitivity (tpr), [56](#)
smape, [9](#), [23](#), [25](#), [26](#), [29](#), [30](#), [34–37](#), [39](#), [42–44](#),
[46–49](#), [49](#), [51](#), [52](#)
specificity (tnr), [54](#)
srho, [9](#), [23](#), [25](#), [26](#), [29](#), [30](#), [34–37](#), [39](#), [42–44](#),
[46–50](#), [51](#), [52](#)
sse, [9](#), [23](#), [25](#), [26](#), [29](#), [30](#), [34–37](#), [39](#), [42–44](#),
[46–51](#), [52](#)
stats::cor(), [22](#), [51](#)

tn, [5](#), [8](#), [12](#), [14–16](#), [18–20](#), [22](#), [32](#), [38](#), [41](#), [53](#),
[55–57](#)
tnr, [5](#), [8](#), [12](#), [14–16](#), [18–20](#), [22](#), [32](#), [38](#), [41](#), [53](#),
[54](#), [56](#), [57](#)
tp, [5](#), [8](#), [12](#), [14–16](#), [18–20](#), [22](#), [32](#), [38](#), [41](#), [53](#),
[55](#), [55](#), [57](#)
tpr, [5](#), [8](#), [12](#), [14–16](#), [18–20](#), [22](#), [32](#), [38](#), [41](#), [53](#),
[55](#), [56](#), [56](#)