

# Package ‘mlr3measures’

October 5, 2020

**Title** Performance Measures for ‘mlr3’

**Version** 0.3.0

**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, PRROC

**Suggests** testthat

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Collate** 'assertions.R' 'bibentries.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\_prauc.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\_rsqr.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** 2020-10-05 14:10:03 UTC

## R topics documented:

mlr3measures-package . . . . .	3
acc . . . . .	4
auc . . . . .	5
bacc . . . . .	6
bbrier . . . . .	7
bias . . . . .	9
ce . . . . .	10
confusion_matrix . . . . .	11
dor . . . . .	12
fbeta . . . . .	13
fdr . . . . .	15
fn . . . . .	16
fnr . . . . .	17
fomr . . . . .	19
fp . . . . .	20
fpr . . . . .	21
ktau . . . . .	23
logloss . . . . .	24
mae . . . . .	25
mape . . . . .	26
mauc_aunu . . . . .	27
maxae . . . . .	29
maxse . . . . .	30
mbrier . . . . .	31
mcc . . . . .	32
measures . . . . .	33
medae . . . . .	34
medse . . . . .	35
mse . . . . .	36
msle . . . . .	37
npv . . . . .	38
pbias . . . . .	39
ppv . . . . .	40
prauc . . . . .	42
rae . . . . .	43
rmse . . . . .	44
rmsle . . . . .	45
rrse . . . . .	46
rse . . . . .	48

<i>mlr3measures-package</i>	3
rsq . . . . .	49
sae . . . . .	50
smape . . . . .	51
srho . . . . .	52
sse . . . . .	53
tn . . . . .	54
tnr . . . . .	55
tp . . . . .	56
tpr . . . . .	58
<b>Index</b>	<b>60</b>

---

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

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.
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: 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::aidcnr2820030106>3.0.co;23](https://doi.org/10.1002/1097-0142(1950)3:1<32::aidcnr2820030106>3.0.co;23).

**See Also**

Other Binary Classification Measures: `bbrier()`, `dor()`, `fbeta()`, `fdr()`, `fnr()`, `fn()`, `fomr()`, `fpr()`, `fp()`, `mcc()`, `npv()`, `ppv()`, `prauc()`, `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**

<code>truth</code>	:: <code>factor()</code> True (observed) labels. Must have the same levels and length as <code>response</code> .
<code>response</code>	:: <code>factor()</code> Predicted response labels. Must have the same levels and length as <code>truth</code> .
<code>sample_weights</code>	:: <code>numeric()</code> Non-negative sample weights. Must have the same levels and length as <code>truth</code> . Defaults to equal sample weights.
<code>...</code>	:: 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](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\(\)](#), [prauc\(\)](#), [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	<i>Calculate Binary Confusion Matrix</i>
------------------	--

---

### 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**

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:  $[0, \infty)$
- 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](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()`, `prauc()`, `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

*F-beta Score*

---

## 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  $\beta$  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

<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>beta</code>	:: <code>numeric(1)</code> 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**

Rijsbergen, Van CJ (1979). *Information Retrieval*, 2nd edition. Butterworth-Heinemann, Newton, MA, USA. ISBN 408709294. Sasaki, Yutaka, 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-260ct07.pdf>.

**See Also**

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

*False Negative Rate*


---

**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](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()`, `prauc()`, `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)
fnr(truth, response, positive = "a")

```

---

fomr	<i>False Omission Rate</i>
------	----------------------------

---

**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](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\(\)](#), [prauc\(\)](#), [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	<i>False Positives</i>
----	------------------------

---

## 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

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](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\(\)](#), [prauc\(\)](#), [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](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\(\)](#), [prauc\(\)](#), [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	<i>Kendall's tau</i>
------	----------------------

---

### 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

*Log Loss*


---

**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**

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

**Note**

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

**References**

de Myttenaere, Arnaud, Golden, Boris, Le Grand, Bénédicte, Rossi, Fabrice (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_aulu(truth, prob, na_value = NaN, ...)
```

```
mauc_aulp(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, Tom (2001). "Using rule sets to maximize ROC performance." In *Proceedings 2001 IEEE international conference on data mining*, 131–138. IEEE. Ferri, César, Hernández-Orallo, José, 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, J D, Till, J R (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**

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\(\)](#), [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	<i>Max Squared Error</i>
-------	--------------------------

---

**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\(\)](#), [prauc\(\)](#), [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 52.

**Examples**

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

---

<code>medae</code>	<i>Median Absolute Errors</i>
--------------------	-------------------------------

---

**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 response.
<code>response</code>	<code>:: numeric()</code>	Predicted response values. Must have the same length as truth.
<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

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

### 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	<i>Mean Squared Log Error</i>
------	-------------------------------

---

**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**

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  $FN + TN = 0$ .

**References**

[https://en.wikipedia.org/wiki/Template:DiagnosticTesting\\_Diagram](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\(\)](#), [prauc\(\)](#), [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](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()`, `prauc()`, `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")
```

---

prauc

*Area Under the Precision-Recall Curve*

---

### Description

Computes the area under the Precision-Recall curve (PRC). The PRC can be interpreted as the relationship between precision and recall (sensitivity), and is considered to be a more appropriate measure for unbalanced datasets than the ROC curve. The PRC is computed by integration of the piecewise function.

### Usage

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

### 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.
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: prob

### Note

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

## References

Davis J, Goadrich M (2006). “The relationship between precision-recall and ROC curves.” In *Proceedings of the 23rd International Conference on Machine Learning*. ISBN 9781595933836.

## See Also

Other Binary Classification Measures: [auc\(\)](#), [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)
prauc(truth, prob, "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

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\(\)](#), [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)
rmsle(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**

<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 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

<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\(\)](#), [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**

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, 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

*Spearman's rho*


---

**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**

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()`, `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](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\(\)](#), [prauc\(\)](#), [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	<i>True Negative Rate</i>
-----	---------------------------

---

**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](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\(\)](#), [prauc\(\)](#), [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

*True Positives*

---

**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](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\(\)](#), [prauc\(\)](#), [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](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\(\)](#), [prauc\(\)](#), [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

## \* Binary Classification Measures

auc, 5  
bbrier, 7  
dor, 12  
fbeta, 13  
fdr, 15  
fn, 16  
fnr, 17  
fomr, 19  
fp, 20  
fpr, 21  
mcc, 32  
npv, 38  
ppv, 40  
prauc, 42  
tn, 54  
tnr, 55  
tp, 56  
tpr, 58

## \* Classification Measures

acc, 4  
bacc, 6  
ce, 10  
logloss, 24  
mauc\_aunu, 27  
mbrier, 31

## \* Regression Measures

bias, 9  
ktau, 23  
mae, 25  
mape, 26  
maxae, 29  
maxse, 30  
medae, 34  
medse, 35  
mse, 36  
msle, 37  
pbias, 39  
rae, 43

rmse, 44  
rmsle, 45  
rrse, 46  
rse, 48  
rsq, 49  
sae, 50  
smape, 51  
srho, 52  
sse, 53

## \* binary\_classification\_measure

auc, 5  
bbrier, 7  
dor, 12  
fbeta, 13  
fdr, 15  
fn, 16  
fnr, 17  
fomr, 19  
fp, 20  
fpr, 21  
mcc, 32  
npv, 38  
ppv, 40  
prauc, 42  
tn, 54  
tnr, 55  
tp, 56  
tpr, 58

## \* classification\_measure

acc, 4  
bacc, 6  
ce, 10  
logloss, 24  
mauc\_aunu, 27  
mbrier, 31

## \* datasets

measures, 33

## \* regression\_measure

bias, 9

- ktau, 23
- mae, 25
- mape, 26
- maxae, 29
- maxse, 30
- medae, 34
- medse, 35
- mse, 36
- msle, 37
- pbias, 39
- rae, 43
- rmse, 44
- rmsle, 45
- rrse, 46
- rse, 48
- rsq, 49
- sae, 50
- smape, 51
- srho, 52
- sse, 53
  
- acc, 4, 7, 10, 25, 28, 32
- auc, 5, 8, 13, 14, 16–18, 20–22, 33, 39, 41, 43, 55–57, 59
  
- bacc, 4, 6, 10, 25, 28, 32
- bbrier, 6, 7, 13, 14, 16–18, 20–22, 33, 39, 41, 43, 55–57, 59
- bbrier(), 31
- bias, 9, 23, 26, 27, 29, 30, 34–37, 40, 44–48, 50–54
  
- ce, 4, 7, 10, 25, 28, 32
- confusion\_matrix, 11
  
- dor, 6, 8, 12, 14, 16–18, 20–22, 33, 39, 41, 43, 55–57, 59
  
- environment(), 33
  
- fbeta, 6, 8, 13, 13, 16–18, 20–22, 33, 39, 41, 43, 55–57, 59
- fdr, 6, 8, 13, 14, 15, 17, 18, 20–22, 33, 39, 41, 43, 55–57, 59
- fn, 6, 8, 13, 14, 16, 16, 18, 20–22, 33, 39, 41, 43, 55–57, 59
- fnr, 6, 8, 13, 14, 16, 17, 17, 20–22, 33, 39, 41, 43, 55–57, 59
- fomr, 6, 8, 13, 14, 16–18, 19, 21, 22, 33, 39, 41, 43, 55–57, 59
  
- fp, 6, 8, 13, 14, 16–18, 20, 20, 22, 33, 39, 41, 43, 55–57, 59
- fpr, 6, 8, 13, 14, 16–18, 20, 21, 21, 33, 39, 41, 43, 55–57, 59
  
- ktau, 9, 23, 26, 27, 29, 30, 34–37, 40, 44–48, 50–54
  
- logloss, 4, 7, 10, 24, 28, 32
  
- mae, 9, 23, 25, 27, 29, 30, 34–37, 40, 44–48, 50–54
- mape, 9, 23, 26, 26, 29, 30, 34–37, 40, 44–48, 50–54
- mauc\_au1p (mauc\_aunu), 27
- mauc\_au1u (mauc\_aunu), 27
- mauc\_aunp (mauc\_aunu), 27
- mauc\_aunu, 4, 7, 10, 25, 27, 32
- maxae, 9, 23, 26, 27, 29, 30, 34–37, 40, 44–48, 50–54
- maxse, 9, 23, 26, 27, 29, 30, 34–37, 40, 44–48, 50–54
- mbrier, 4, 7, 10, 25, 28, 31
- mbrier(), 7
- mcc, 6, 8, 13, 14, 16–18, 20–22, 32, 39, 41, 43, 55–57, 59
- measures, 33
- medae, 9, 23, 26, 27, 29, 30, 34, 35–37, 40, 44–48, 50–54
- medse, 9, 23, 26, 27, 29, 30, 34, 35, 36, 37, 40, 44–48, 50–54
- mlr3measures (mlr3measures-package), 3
- mlr3measures-package, 3
- mse, 9, 23, 26, 27, 29, 30, 34, 35, 36, 37, 40, 44–48, 50–54
- msle, 9, 23, 26, 27, 29, 30, 34–36, 37, 40, 44–48, 50–54
  
- npv, 6, 8, 13, 14, 16–18, 20–22, 33, 38, 41, 43, 55–57, 59
  
- pbias, 9, 23, 26, 27, 29, 30, 34–37, 39, 44–48, 50–54
- ppv, 6, 8, 13, 14, 16–18, 20–22, 33, 39, 40, 43, 55–57, 59
- prauc, 6, 8, 13, 14, 16–18, 20–22, 33, 39, 41, 42, 55–57, 59
- precision, 14
- precision (ppv), 40

precision(), 13

rae, 9, 23, 26, 27, 29, 30, 34–37, 40, 43,  
45–48, 50–54

recall, 14

recall(tpr), 58

recall(), 13

rmse, 9, 23, 26, 27, 29, 30, 34–37, 40, 44, 44,  
46–48, 50–54

rmsle, 9, 23, 26, 27, 29, 30, 34–37, 40, 44, 45,  
45, 47, 48, 50–54

rrse, 9, 23, 26, 27, 29, 30, 34–37, 40, 44–46,  
46, 48, 50–54

rse, 9, 23, 26, 27, 29, 30, 34–37, 40, 44–47,  
48, 50–54

rse(), 49

rsq, 9, 23, 26, 27, 29, 30, 34–37, 40, 44–48,  
49, 51–54

sae, 9, 23, 26, 27, 29, 30, 34–37, 40, 44–48,  
50, 50, 52–54

sensitivity(tpr), 58

smape, 9, 23, 26, 27, 29, 30, 34–37, 40, 44–48,  
50, 51, 51, 53, 54

specificity(tnr), 55

srho, 9, 23, 26, 27, 29, 30, 34–37, 40, 44–48,  
50–52, 52, 54

sse, 9, 23, 26, 27, 29, 30, 34–37, 40, 44–48,  
50–53, 53

stats::cor(), 23, 52

tn, 6, 8, 13, 14, 16–18, 20–22, 33, 39, 41, 43,  
54, 56, 57, 59

tnr, 6, 8, 13, 14, 16–18, 20–22, 33, 39, 41, 43,  
55, 55, 57, 59

tp, 6, 8, 13, 14, 16–18, 20–22, 33, 39, 41, 43,  
55, 56, 56, 59

tpr, 6, 8, 13, 14, 16–18, 20–22, 33, 39, 41, 43,  
55–57, 58