Package ‘raptools’

May 9, 2026

Title Risk Assessment Plot and Reclassification Metrics

Version 1.23.0

Description Assessing the comparative performance of two logistic regression models or results of such models or classification models. Discrimination metrics include Integrated Discrimination Improvement (IDI), Net Reclassification Improvement (NRI), and difference in Area Under the Curves (AUCs), Brier scores and Brier skill. Plots include Risk Assessment Plots, Decision curves and Calibration plots. Methods are described in Pickering and Endre (2012) <doi:10.1373/clinchem.2011.167965> and Pencina et al. (2008) <doi:10.1002/sim.2929>.

Depends R (>= 4.1.0)

Imports rms, Hmisc, dplyr, ggplot2, pROC, stats, tidyr, forcats, pracma, ggrepel

License GPL-3

LazyData true

LazyLoad true

RoxygenNote 7.3.2

Encoding UTF-8

URL `https://github.com/Researchverse/raptools`,
`https://researchverse.github.io/raptools/`

BugReports `https://github.com/Researchverse/raptools/issues`

NeedsCompilation no

Author John W Pickering [aut],
Dimitrios Doudesis [aut],
Daniel Perez Vicencio [cre]

Maintainer Daniel Perez Vicencio <`dvicencio947@gmail.com`>

Repository CRAN

Date/Publication 2025-12-09 21:50:13 UTC

Contents

`anova_glm` *The function anova_glm() returns the Chi^2 and degrees of freedom for each variable & the same was anova.rms() does from lrm() in the rms package.*

Description

The function anova_glm() returns the Chi^2 and degrees of freedom for each variable & the same was anova.rms() does from lrm() in the rms package.

Usage

```
anova_glm(f)
```

Arguments

`f`	A logistic regression fit created using glm (base package)

Value

A data frame with Chi-Square values and degrees of freedom for each variable in the model, plus a TOTAL row summarizing the overall model statistics.

`CI.classNRI` *Statistical metrics and confidence intervals for classes*

Description

The function CI.classNRI calculates the NRI statistics for reclassification of data already in classes with confidence intervals. Uses statistics.classNRI.

Usage

```
CI.classNRI(
  c1,
  c2,
  y,
  s1 = NULL,
  s2 = NULL,
  conf.level = 0.95,
  n.boot = 1000,
  dp = 3
)
```

Arguments

`c1`	Risk classes of the baseline model (ordinal)
`c2`	Risk classes of new model
`y`	Binary of outcome of interest. Must be 0 or 1.
`s1`	The savings or benefit when am event is reclassified to a higher group by the new model (positive numeric)
`s2`	The benefit when a non-event is reclassified to a lower group (positive numeric)
`conf.level`	The confidence interval expressed as a fraction of 1 (ie 0.95 is the 95% confidence interval)
`n.boot`	The number of "bootstraps" to use. Performance slows down with more bootstraps. For trialling result, use a low number (eg 2), for accuracy use a large number (eg 2000)
`dp`	The number of decimal places to display

Value

A list with the following elements:

meta_data Some overall meta data - Confidence Interval, number of bootstraps, s1, s2

Metrics Point estimates of the statistical metrics.

Each_bootstrap_metrics Point estimates of the statistical metrics for each bootstrapped sample.

Summary_metrics Point estimates with confidence intervals of the statistical metrics (e.g. Total, Events, Non-events, Prevalence, NRI, IDI, confusion matrices).

A matrix of metrics

`CI.raplot` *Statistical metrics with confidence intervals*

Description

The CI.raplot function produces summary metrics for risk assessment. Outputs the NRI, IDI, weighted NRI and category Free NRI all for those with events and those without events. Also the AUCs of the two models and the comparison (DeLong) between AUCs. Output includes confidence intervals. Uses statistics.raplot. Displayed graphically by raplot.

Usage

```
CI.raplot(
  x1,
  x2 = NULL,
  y = NULL,
  t = NULL,
  NRI_return = FALSE,
  conf.level = 0.95,
  n.boot = 1000,
  dp = 3
)
```

Arguments

`x1`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the baseline model. Must be between 0 & 1
`x2`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the new (alternative) model. Must be between 0 & 1
`y`	Binary of outcome of interest. Must be 0 or 1 (if fitted models are provided this is extracted from the fit which for an rms fit must have x = TRUE, y = TRUE).
`t`	The risk threshold(s) for groups. eg t<-c(0,0.1,1) is a two group model with a threshold of 0.1 & t<-c(0,0.1,0.3,1) is a three group model with thresholds at 0.1 and 0.3.
`NRI_return`	If NRI statistics are required (default = FALSE).
`conf.level`	The confidence interval expressed as a fraction of 1 (ie 0.95 is the 95% confidence interval)
`n.boot`	The number of "bootstraps" to use. Performance slows down with more bootstraps. For trialling result, use a low number (eg 5), for accuracy use a large number (eg 2000)
`dp`	The number of decimal places to display

Value

A list with the following elements:

meta_data A data.frame with thresholds, confidence interval, number of bootstraps, input data type and decimal places.

Metrics Point estimates of the statistical metrics (see function docs).

Each_bootstrap_metrics List of per-bootstrap metric results.

Summary_metrics A table of summary metrics with confidence intervals (e.g. Total, Events, Non-events, NRI, IDI, AUCs, Brier scores, etc.).

References

Pencina, M. J., D'Agostino, R. B., & Vasan, R. S. (2008). Evaluating the added stats::predictive ability of a new marker: From area under the ROC curve to reclassification and beyond. Statistics in Medicine, 27(2), 157-172. doi:10.1002/sim.2929

Examples

```
# Quick example with subset of data and fewer bootstraps
data(data_risk)
data_subset <- data_risk[1:100, ]  # Use first 100 rows for speed
complete_cases <- complete.cases(data_subset)
data_clean <- data_subset[complete_cases, ]
y <- data_clean$outcome
x1 <- data_clean$baseline
x2 <- data_clean$new
t <- c(0, 0.19, 1)
output <- CI.raplot(x1, x2, y, t, conf.level = 0.95, n.boot = 10, dp = 2)


# Full dataset example with more bootstraps
data(data_risk)
complete_cases <- complete.cases(data_risk)
data_clean <- data_risk[complete_cases, ]
y <- data_clean$outcome
x1 <- data_clean$baseline
x2 <- data_clean$new
t <- c(0, 0.19, 1)
output <- CI.raplot(x1, x2, y, t, conf.level = 0.95, n.boot = 1000, dp = 2)
```

`data_class` *Simple data set with classifications*

Description

Example data for use with CI.classNRI

Usage

```
data_class
```

Format

data frame with 3 columns

ref_class The class of the baseline model. Must be a factor

new_class The class of the new model. Must be a factor

Outcome The outcome of interest (Low or High). Must be a factor

---

`data_risk` *Simple data set with risk predictions*

Description

Example data for use with CI.raplot

Usage

```
data_risk
```

Format

data frame with 3 columns

ref The prediction from the baseline model

new The prediction from the new model

outcome The outcome of interest (0 or 1)

`extractCI` *Extract confidence interval*

Description

Extract a confidence in interval from the bootstrapped results. Used by CI.raplot

Usage

```
extractCI(results.boot, conf.level, n.boot, dp)
```

Arguments

`results.boot`	The matrix of n.boot metrics from within CI.raplot
`conf.level`	The confidence interval expressed between 0 & 1 (eg 95%CI is conf.level = 0.95)
`n.boot`	The number of bootstrapped samples
`dp`	the number of decimal places to report the point estimate and confidence interval

Value

A two column matrix with the metric name and statistic with a confidence interval

---

`extract_NRI_CI` *Extract NRI confidence intervals*

Description

Extract a confidence in interval from the bootstrapped results. Used by CI.NRI

Usage

```
extract_NRI_CI(results.boot, conf.level, n.boot, dp)
```

Arguments

`results.boot`	The matrix of n.boot metrics from within CI.NRI
`conf.level`	The confidence interval expressed between 0 & 1 (eg 95%CI is conf.level = 0.95)
`n.boot`	The number of bootstrapped samples
`dp`	the number of decimal places to report the point estimate and confidence interval

Value

A two column matrix with the metric name and statistic with a confidence interval

`ggcalibrate` *The Calibration plot*

Description

ggcalibrate plots the stats::predicted events against the actual event rate

Usage

```
ggcalibrate(
  x1,
  x2 = NULL,
  y = NULL,
  n_knots = 5,
  ci_level = 0.95,
  smooth_method = "loess",
  smooth_span = 0.75
)
```

Arguments

`x1`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the baseline model. Must be between 0 & 1
`x2`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the new (alternative) model. Must be between 0 & 1
`y`	Binary of outcome of interest. Must be 0 or 1 (if fitted models are provided this is extracted from the fit which for an rms fit must have x = TRUE, y = TRUE).
`n_knots`	The curves are made by fitting a restricted cubic spline (rms package). The default 5-knots is usually enough.
`ci_level`	Confidence interval of the curve (default = 0.95).
`smooth_method`	Smoothing method for geom_smooth. Options: "loess", "lm", "glm", "gam". Default is "loess"
`smooth_span`	Span parameter for loess smoothing, controls the degree of smoothing (default = 0.75). Lower values = less smooth

Value

a ggplot

Examples

```
# Quick example with subset of data
data(data_risk)
data_subset <- data_risk[1:100, ]  # Use first 100 rows for speed
complete_cases <- complete.cases(data_subset)
data_clean <- data_subset[complete_cases, ]
y <- data_clean$outcome
x1 <- data_clean$baseline
x2 <- data_clean$new
output <- ggcalibrate(x1, x2, y, n_knots = 3, ci_level = 0.95)


# Full dataset example
data(data_risk)
complete_cases <- complete.cases(data_risk)
data_clean <- data_risk[complete_cases, ]
y <- data_clean$outcome
x1 <- data_clean$baseline
x2 <- data_clean$new
output <- ggcalibrate(x1, x2, y, n_knots = 5, ci_level = 0.95)
```

`ggcalibrate_original` *The Original Calibration plot*

Description

ggcalibrate_original plots the stats::predicted events against the actual event rate using the "old" form.

Usage

```
ggcalibrate_original(
  x1,
  x2 = NULL,
  y = NULL,
  n_cut = 5,
  cut_type = c("interval", "number", "width"),
  include_margin = FALSE
)
```

Arguments

`x1`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the baseline model. Must be between 0 & 1
`x2`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the new (alternative) model. Must be between 0 & 1

`y`	Binary of outcome of interest. Must be 0 or 1 (if fitted models are provided this is extracted from the fit which for an rms fit must have x = TRUE, y = TRUE).
`n_cut`	An integer indicating either the number of intervals of the same width, the number of intervals of the same number of subjects, or the width (as a percentage) of the intervals.
`cut_type`	One of three strings: "interval", "number", or "width". - "interval": uses cut_interval() to get n_cut intervals of approximately equal width. - "number": uses cut_number() to get n_cut intervals with approximately equal counts. - "width": uses cut_width() to get intervals of a fixed width (approximately 100/n_cut).
`include_margin`	TRUE for including producing a bar plot of the counts of in each of the intervals. Default is FALSE. Note if the output is saved to my_graphs then using the library gridExtra the function grid.arrange(graphs$g, graphs$g_marg , nrow = 2, heights = c(2,1)) will produce a plot with both the calibration plot and the marginal plot.

Value

a list of one or two ggplots

Examples

```
# Quick example with subset of data
data(data_risk)
data_subset <- data_risk[1:100, ]  # Use first 100 rows for speed
complete_cases <- complete.cases(data_subset)
data_clean <- data_subset[complete_cases, ]
y <- data_clean$outcome
x1 <- data_clean$baseline
x2 <- data_clean$new
output <- ggcalibrate_original(
  x1, x2, y,
  n_cut = 3, cut_type = "interval",
  include_margin = FALSE
)


# Full dataset example
data(data_risk)
complete_cases <- complete.cases(data_risk)
data_clean <- data_risk[complete_cases, ]
y <- data_clean$outcome
x1 <- data_clean$baseline
x2 <- data_clean$new
output <- ggcalibrate_original(
  x1, x2, y,
  n_cut = 5, cut_type = "interval",
  include_margin = FALSE
)
```

`ggcontribute` *The Contribution plot*

Description

ggcontribute plots the contribution of each variable to the model

Usage

```
ggcontribute(x1, x2 = NULL, option_flag = c("chi2", "percent"))
```

Arguments

`x1`	Either a logistic regression fitted using glm (base package) or lrm (rms package) of the baseline model.
`x2`	Either a logistic regression fitted using glm (base package) or lrm (rms package) of the new (alternative) model.
`option_flag`	A flag to choose if the relative percentage of the Chi2-degrees of freedom are plotted.

Value

A ggplot object displaying the contribution of each variable to the model(s) using either Chi-square minus degrees of freedom or relative percentage contribution. If two models are provided, arrows show the change in contribution between models.

`ggdecision` *The Decision curve*

Description

ggdecision plots decision curves to assess the net benefit at different thresholds

ggdecision plots decision curves to assess the net benefit at different thresholds

Usage

```
ggdecision(
  x1,
  x2 = NULL,
  y = NULL,
  show_smooth = TRUE,
  smooth_method = "loess",
  smooth_span = 0.75,
  smooth_se = FALSE
)
```

```
ggdecision(
  x1,
  x2 = NULL,
  y = NULL,
  show_smooth = TRUE,
  smooth_method = "loess",
  smooth_span = 0.75,
  smooth_se = FALSE
)
```

Arguments

`x1`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the baseline model. Must be between 0 & 1
`x2`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the new (alternative) model. Must be between 0 & 1
`y`	Binary of outcome of interest. Must be 0 or 1 (if fitted models are provided this is extracted from the fit which for an rms fit must have x = TRUE, y = TRUE).
`show_smooth`	Logical, whether to display smoothed curves (default = TRUE)
`smooth_method`	Smoothing method for geom_smooth. Options: "loess", "lm", "glm", "gam". Default is "loess"
`smooth_span`	Span parameter for loess smoothing, controls the degree of smoothing (default = 0.75). Lower values = less smooth
`smooth_se`	Logical, whether to display confidence interval around smooth (default = FALSE)

Value

a ggplot

a ggplot

References

Vickers AJ, van Calster B, Steyerberg EW. A simple, step-by-step guide to interpreting decision curve analysis. Diagn Progn Res 2019;3(1):18. 2. Zhang Z, Rousson V, Lee W-C, et al. Decision curve analysis: a technical note. Ann Transl Med 2018;6(15):308-308.

Vickers AJ, van Calster B, Steyerberg EW. A simple, step-by-step guide to interpreting decision curve analysis. Diagn Progn Res 2019;3(1):18. 2. Zhang Z, Rousson V, Lee W-C, et al. Decision curve analysis: a technical note. Ann Transl Med 2018;6(15):308–308.

`ggprerec` *The Precision-Recall plot*

Description

ggprerec plots Precision (PPV) v Recall (Sensitivity)

Usage

```
ggprerec(
  x1,
  x2 = NULL,
  y = NULL,
  show_smooth = TRUE,
  smooth_method = "loess",
  smooth_span = 0.75,
  smooth_se = FALSE
)
```

Arguments

`x1`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or alculated probabilities (eg through a logistic regression model) of the baseline model. Must be between 0 & 1
`x2`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the new (alternative) model. Must be between 0 & 1
`y`	Binary of outcome of interest. Must be 0 or 1 (if fitted models are provided this is extracted from the fit which for an rms fit must have x = TRUE, y = TRUE).
`show_smooth`	Logical, whether to display smoothed curves (default = TRUE)
`smooth_method`	Smoothing method for geom_smooth. Options: "loess", "lm", "glm", "gam". Default is "loess"
`smooth_span`	Span parameter for loess smoothing, controls the degree of smoothing (default = 0.75). Lower values = less smooth
`smooth_se`	Logical, whether to display confidence interval around smooth (default = FALSE)

Value

A ggplot object displaying the precision-recall curve(s) with recall (sensitivity) on the x-axis and precision (positive predictive value) on the y-axis. If two models are provided, both curves are shown for comparison.

`ggrap` *The Risk Assessment Plot*

Description

The function ggrap() plots the Sensitivity and 1-Specificity curves against the calculated risk for the baseline (reference) and newmodels, thus graphically displaying the IDIs for those with and without the events. These plots can aid interpretation of the NRI and IDI metrics.

Usage

```
ggrap(x1, x2 = NULL, y = NULL)
```

```
ggrap(x1, x2 = NULL, y = NULL)
```

Arguments

`x1`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or alculated probabilities (eg through a logistic regression model) of the baseline model. Must be between 0 & 1
`x2`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the new (alternative) model. Must be between 0 & 1
`y`	Binary of outcome of interest. Must be 0 or 1 (if fitted models are provided this is extracted from the fit which for an rms fit must have x = TRUE, y = TRUE).

Value

a ggplot

a ggplot

References

The Risk Assessment Plot in this form was described by Pickering, J. W., & Endre, Z. H. (2012). New Metrics for Assessing Diagnostic Potential of Candidate Biomarkers. Clinical Journal of the American Society of Nephrology, 7, 1355–1364. doi:10.2215/CJN.09590911

The Risk Assessment Plot in this form was described by Pickering, J. W., & Endre, Z. H. (2012). New Metrics for Assessing Diagnostic Potential of Candidate Biomarkers. Clinical Journal of the American Society of Nephrology, 7, 1355–1364. doi:10.2215/CJN.09590911

`ggroc` *The ROC plot*

Description

ggroc plots Sensitivity v 1-Specificity

Usage

```
ggroc(
  x1,
  x2 = NULL,
  y = NULL,
  carrington_line = FALSE,
  costs = c(0, 0, 1, 1),
  label_number = NULL
)
```

Arguments

`x1`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or alculated probabilities (eg through a logistic regression model) of the baseline model. Must be between 0 & 1
`x2`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the new (alternative) model. Must be between 0 & 1
`y`	Binary of outcome of interest. Must be 0 or 1 (if fitted models are provided this is extracted from the fit which for an rms fit must have x = TRUE, y = TRUE).
`carrington_line`	The Useful Area is from the roc down to this line. It depends on prevalence and the costs of FP, FN, TP, TN. Default is FALSE. See Carrington et al.
`costs`	Numeric vectors costs = c(cFP, cFN,cTP, cTN). The costs of FP, FN, TP, TN. Default, c(0,0,1,1), is for there to be no costs for the FP & FN and identical costs for TN and TP. See Carrington et al.
`label_number`	The number of points on the curve to label.The default has no labels.

Value

A ggplot object displaying the ROC curve(s) with sensitivity on the y-axis and 1-specificity on the x-axis. If two models are provided, both curves are shown for comparison.

References

Carrington AM, Fieguth PW, Mayr F, James ND, Holzinger A, Pickering JW, et al. The ROC Diagonal is not Layperson's Chance: a New Baseline Shows the Useful Area. Machine Learning and Knowledge Extraction. Vienna, Austria: Springer; 2022. pp. 100-113. Available: 10.1007/978-3-031-14463-9_7.

`meta.rap` *List meta data*

Description

Display the meta data

Usage

```
meta.rap(l)
```

Arguments

`l`	List returned from CI.raplot

Value

A tibble

`statistics.classNRI` *Reclassification metrics with classes (ordinals) as inputs*

Description

The function statistics.classNRI calculates the NRI metrics for reclassification of data already in classes. For use by CI.classNRI.

Usage

```
statistics.classNRI(c1, c2, y, s1 = NULL, s2 = NULL)
```

Arguments

`c1`	Risk class of Reference model (ordinal factor).
`c2`	Risk class of New model (ordinal factor)
`y`	Binary of outcome of interest. Must be 0 or 1.
`s1`	The savings or benefit when an event is reclassified to a higher group by the new model. i.e instead of counting as 1 an event classified to a higher group, it is counted as s1.
`s2`	The benefit when a non-event is reclassified to a lower group. i.e instead of counting as 1 an event classified to a lower group, it is counted as s2.

Value

A matrix of metrics for use within CI.classNRI

Examples

```
# Quick example
data(data_class)
data_subset <- data_class[1:100, ]  # Use first 100 rows for speed
y <- data_subset$outcome
c1 <- data_subset$base_class
c2 <- data_subset$new_class
output <- statistics.classNRI(c1, c2, y)


# Full dataset example
data(data_class)
y <- data_class$outcome
c1 <- data_class$base_class
c2 <- data_class$new_class
output <- statistics.classNRI(c1, c2, y)
```

`statistics.raplot` *Statistical metrics*

Description

The function statistics.raplot calculates the reclassification metrics. Used by CI.raplot.

Usage

```
statistics.raplot(x1, x2, y, t = NULL, NRI_return = FALSE)
```

Arguments

`x1`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the baseline model. Must be between 0 & 1
`x2`	Either a logistic regression fitted using glm (base package) or lrm (rms package) or calculated probabilities (eg through a logistic regression model) of the new (alternative) model. Must be between 0 & 1
`y`	Binary of outcome of interest. Must be 0 or 1 (if fitted models are provided this is extracted from the fit which for an rms fit must have x = TRUE, y = TRUE).
`t`	The risk threshold(s) for groups. eg t<-c(0,0.1,1) is a two group scenario with a threshold of 0.1 & t<-c(0,0.1,0.3,1) is a three group scenario with thresholds at 0.1 and 0.3. Nb. If no t is provided it defaults to a single threshold at the prevalence of the cohort.
`NRI_return`	Flag to return NRI metrics, default is FALSE.

Value

A matrix of metrics for use within CI.raplot

`summary.rap` *List risk assessment metrics*

Description

Display the summary metrics

Usage

```
## S3 method for class 'rap'
summary(l)
```

Arguments

`l`	List returned from CI.raplot

Value

A tibble

Index