Package ‘CalibrationCurves’

August 13, 2022

Type Package

Title Calibration Performance

Version 0.1.5

Date 2022-08-11

Description Plots calibration curves and computes statistics for assessing calibration performance. See Van Calster et al. (2016) <doi:10.1016/j.jclinepi.2015.12.005>.

License GPL (>= 3)

Depends rms

Imports grDevices, graphics, methods, stats, utils, survival, Hmisc

RoxygenNote 7.2.0

Encoding UTF-8

NeedsCompilation no

Author De Cock Bavo [aut, cre],
Nieboer Daan [aut],
Van Calster Ben [aut],
Steyerberg Ewout [aut],
Vergouwe Yvonne [aut]

Maintainer De Cock Bavo <bavo.decock@kuleuven.be>

Repository CRAN

Date/Publication 2022-08-13 12:50:06 UTC

R topics documented:

`.rcspline.plot` *Internal function*

Description

Adjusted version of the `rcspline.plot` function where only the output is returned and no plot is made

Usage

```
.rcspline.plot(
  x,
  y,
  model = c("logistic", "cox", "ols"),
  xrange,
  event,
  nk = 5,
  knots = NULL,
  show = c("xbeta", "prob"),
  adj = NULL,
  xlab,
  ylab,
  ylim,
  plim = c(0, 1),
  plotcl = TRUE,
  showknots = TRUE,
  add = FALSE,
  subset,
  lty = 1,
  noprint = FALSE,
  m,
  smooth = FALSE,
  bass = 1,
  main = "auto",
  statloc
)
```

Arguments

`x` a numeric predictor

`y` a numeric response. For binary logistic regression, `y` should be either 0 or 1.

`model` `"logistic"` or `"cox"`. For `"cox"`, uses the `coxph.fit` function with `method="efron"` arguement set.

`xrange` range for evaluating `x`, default is f and $1-f$ quantiles of `x`, where $f = \frac{10}{\max{(n, 200)}}$ and n the number of observations `event` event/censoring indicator if `model="cox"`. If `event` is present, `model` is assumed to be `"cox"`

`nk` number of knots

`knots` knot locations, default based on quantiles of `x` (by `rcspline.eval`)

`show` `"xbeta"` or `"prob"` - what is plotted on y-axis

`adj` optional matrix of adjustment variables

`xlab` x-axis label, default is the “label” attribute of `x`

`ylab` y-axis label, default is the “label” attribute of `y`

`ylim` y-axis limits for logit or log hazard

`plim` y-axis limits for probability scale

`plotcl` plot confidence limits

`showknots` show knot locations with arrows

`add` add this plot to an already existing plot

`subset` subset of observations to process, e.g. `sex == "male"`

`lty` line type for plotting estimated spline function

`noprint` suppress printing regression coefficients and standard errors

`m` for `model="logistic"`, plot grouped estimates with triangles. Each group contains `m` ordered observations on `x`.

`smooth` plot nonparametric estimate if `model="logistic"` and `adj` is not specified

`bass` smoothing parameter (see `supsmu`)

`main` main title, default is `"Estimated Spline Transformation"`

`statloc` location of summary statistics. Default positioning by clicking left mouse button where upper left corner of statistics should appear. Alternative is `"ll"` to place below the graph on the lower left, or the actual `x` and `y` coordinates. Use `"none"` to suppress statistics.

Value

list with components (‘`knots`’, ‘`x`’, ‘`xbeta`’, ‘`lower`’, ‘`upper`’) which are respectively the knot locations, design matrix, linear predictor, and lower and upper confidence limits

See Also

`lrm`, `cph`, `rcspline.eval`, `plot`, `supsmu`, `coxph.fit`, `lrm.fit` auc.nonpara.mw *AUC Based on the Mann-Whitney Statistic*

Description

Obtain the point estimate and the confidence interval of the AUC by various methods based on the Mann-Whitney statistic.

Usage

```
auc.nonpara.mw(x, y, conf.level=0.95,
               method=c("newcombe", "pepe", "delong",
                        "jackknife", "bootstrapP", "bootstrapBCa"),
               nboot)
```

Arguments

x	a vector of observations from class P.
y	a vector of observations from class N.
conf.level	confidence level of the interval. The default is 0.95.
method	a method used to construct the CI. newcombe is the method recommended in Newcombe (2006); pepe is the method proposed in Pepe (2003); delong is the method proposed in Delong et al. (1988); jackknife uses the jackknife method; bootstrapP uses the bootstrap with percentile CI; bootstrapBCa uses bootstrap with bias-corrected and accelerated CI. The default is newcombe. It can be abbreviated.
nboot	number of bootstrap iterations.

Details

The function implements various methods based on the Mann-Whitney statistic.

Value

Point estimate and lower and upper bounds of the CI of the AUC.

Note

The observations from class P tend to have larger values then that from class N.

This help-file is a copy of the original help-file of the function auc.nonpara.mw from the auRoc-package. It is important to note that, when using method="pepe", the confidence interval is computed as documented in Qin and Hotilovac (2008) and that this is different from the original function.

References

Elizabeth R Delong, David M Delong, and Daniel L Clarke-Pearson (1988) Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics* **44** 837-845

Dai Feng, Giuliana Cortese, and Richard Baumgartner (2015) A comparison of confidence/credible interval methods for the area under the ROC curve for continuous diagnostic tests with small sample size. *Statistical Methods in Medical Research* DOI: 10.1177/0962280215602040

Robert G Newcombe (2006) Confidence intervals for an effect size measure based on the Mann-Whitney statistic. Part 2: asymptotic methods and evaluation. *Statistics in medicine* **25(4)** 559-573

Margaret Sullivan Pepe (2003) The statistical evaluation of medical tests for classification and prediction. *Oxford University Press*

Qin, G., & Hotilovac, L. (2008). Comparison of non-parametric confidence intervals for the area under the ROC curve of a continuous-scale diagnostic test. *Statistical Methods in Medical Research*, **17(2)**, pp. 207-21

`CalibrationCurves` *General information on package and val.prob.ci.2 function*

Description

Some years ago, Yvonne Vergouwe and Ewout Steyerberg adapted the function `val.prob` from the rms-package (`https://cran.r-project.org/package=rms`) into `val.prob.ci` and added the following functions to `val.prob`:

- Scaled Brier score by relating to max for average calibrated Null model
- Risk distribution according to outcome
- 0 and 1 to indicate outcome label; set with `d1lab=".."`, `d0lab=".."`
- Labels: y axis: "Observed Frequency"; Triangle: "Grouped observations"
- Confidence intervals around triangles
- A cut-off can be plotted; set x coordinate

In December 2015, Bavo De Cock, Daan Nieboer, and Ben Van Calster adapted this to `val.prob.ci.2`:

- Flexible calibration curves can be obtained using loess (default) or restricted cubic splines, with pointwise 95% confidence intervals. Flexible calibration curves are now given by default and this decision is based on the findings reported in Van Calster et al. (2016).
- Loess: confidence intervals can be obtained in closed form or using bootstrapping (CL.BT=T will do bootstrapping with 2000 bootstrap samples, however this will take a while)
- RCS: 3 to 5 knots can be used
  - the knot locations will be estimated using default quantiles of x (by `rcspline.eval`, see `rcspline.plot` and `rcspline.eval`)
  - if estimation problems occur at the specified number of knots (nr.knots, default is 5), the analysis is repeated with nr.knots-1 until the problem has disappeared and the function stops if there is still an estimation problem with 3 knots

- You can now adjust the plot through use of normal plot commands (`cex.axis` etcetera), and the size of the legend now has to be specified in `cex.leg`
- Label y-axis: "Observed proportion"
- Stats: added the Estimated Calibration Index (ECI), a statistical measure to quantify lack of calibration (Van Hoorde et al., 2015)
- Stats to be shown in the plot: by default we show the `"abc"` of model performance (Steyerberg et al., 2011). That is, calibration intercept (calibration-in-the-large), calibration slope and c- statistic. Alternatively, the user can select the statistics of choice (e.g. `dostats=c("C (ROC)","R2")` or `dostats=c(2,3)`.
- Vectors p, y and logit no longer have to be sorted

Since then, several new features have been added and are still being added. The most current version of this package can always be found on `https://github.com/BavoDC` and can easily be installed using the following code: `install.packages("devtools") # if not yet installed require(devtools) install_git("https://github.com/BavoDC/CalibrationCurves")`

References

Steyerberg, E.W.Van Calster, B., Pencina, M.J. (2011). Performance measures for prediction models and markers : evaluation of predictions and classifications. *Revista Espanola de Cardiologia*, **64(9)**, pp. 788-794

Van Calster, B., Nieboer, D., Vergouwe, Y., De Cock, B., Pencina M., Steyerberg E.W. (2016). A calibration hierarchy for risk models was defined: from utopia to empirical data. *Journal of Clinical Epidemiology*, **74**, pp. 167-176

Van Hoorde, K., Van Huffel, S., Timmerman, D., Bourne, T., Van Calster, B. (2015). A spline-based tool to assess and visualize the calibration of multiclass risk predictions. *Journal of Biomedical Informatics*, **54**, pp. 283-93

`print.CalibrationCurve`

Print function for a CalibrationCurve object

Description

Prints the call, confidence level and values for the performance measures.

Usage

```
## S3 method for class 'CalibrationCurve'
print(x, ...)
```

Arguments

`x`	an object of type CalibrationCurve, resulting from `val.prob.ci.2`.
`...`	arguments passed to `print`

Value

The original CalibrationCurve object is returned.

See Also

val.prob.ci.2

---

val.prob.ci.2 *Calibration performance*

Description

The function val.prob.ci.2 is an adaptation of val.prob from Frank Harrell's rms package, https://cran.r-project.org/package=rms. Hence, the description of some of the functions of val.prob.ci.2 come from the the original val.prob.

The key feature of val.prob.ci.2 is the generation of logistic and flexible calibration curves and related statistics. When using this code, please cite: Van Calster, B., Nieboer, D., Vergouwe, Y., De Cock, B., Pencina, M.J., Steyerberg, E.W. (2016). A calibration hierarchy for risk models was defined: from utopia to empirical data. *Journal of Clinical Epidemiology*, **74**, pp. 167-176

Usage

```
val.prob.ci.2(
  p,
  y,
  logit,
  group,
  weights = rep(1, length(y)),
  normwt = FALSE,
  pl = TRUE,
  smooth = c("loess", "rcs", "none"),
  CL.smooth = "fill",
  CL.BT = FALSE,
  lty.smooth = 1,
  col.smooth = "black",
  lwd.smooth = 1,
  nr.knots = 5,
  logistic.cal = FALSE,
  lty.log = 1,
  col.log = "black",
  lwd.log = 1,
  xlab = "Predicted probability",
  ylab = "Observed proportion",
  xlim = c(-0.02, 1),
  ylim = c(-0.15, 1),
  m,
  g,
  cuts,
  emax.lim = c(0, 1),
  legendloc = c(0.5, 0.27),
  statloc = c(0, 0.85),
  dostats = TRUE,
  cl.level = 0.95,
  method.ci = "pepe",
  roundstats = 2,
  riskdist = "predicted",
  cex = 0.75,
  cex.leg = 0.75,
  connect.group = FALSE,
  connect.smooth = TRUE,
  g.group = 4,
  evaluate = 100,
  nmin = 0,
  d0lab = "0",
  d1lab = "1",
  cex.d01 = 0.7,
  dist.label = 0.04,
  line.bins = -0.05,
  dist.label2 = 0.03,
  cutoff,
  las = 1,
  length.seg = 1,
  y.intersp = 1,
  lty.ideal = 1,
  col.ideal = "red",
  lwd.ideal = 1,
  ...
)
```

Arguments

`p` predicted probability

`y` vector of binary outcomes

`logit` predicted log odds of outcome. Specify either `p` or `logit`.

`group` a grouping variable. If numeric this variable is grouped into `g.group` quantile groups (default is quartiles). Set `group=TRUE` to use the `group` algorithm but with a single stratum for `val.prob`.

`weights` an optional numeric vector of per-observation weights (usually frequencies), used only if `group` is given.

`normwt` set to `TRUE` to make `weights` sum to the number of non-missing observations.

`pl` `TRUE` to plot the calibration curve(s). If `FALSE` no calibration curves will be plotted, but statistics will still be computed and outputted.

`smooth` `"loess"` generates a flexible calibration curve based on `loess`, `"rcs"` generates a calibration curves based on restricted cubic splines (see `rcs` and `rcspline.plot`), `"none"` suppresses the flexible curve. We recommend to use loess unless N is large, for example N>5000. Default is `"loess"`.

`CL.smooth` `"fill"` shows pointwise 95% confidence limits for the flexible calibration curve with a gray area between the lower and upper limits, `TRUE` shows pointwise 95% confidence limits for the flexible calibration curve with dashed lines, `FALSE` suppresses the confidence limits. Default is `"fill"`.

`CL.BT` `TRUE` uses confidence limits based on 2000 bootstrap samples, `FALSE` uses closed form confidence limits. Default is `FALSE`.

`lty.smooth` the linetype of the flexible calibration curve. Default is `1`.

`col.smooth` the color of the flexible calibration curve. Default is `"black"`.

`lwd.smooth` the line width of the flexible calibration curve. Default is `1`.

`nr.knots` specifies the number of knots for rcs-based calibration curve. The default as well as the highest allowed value is 5. In case the specified number of knots leads to estimation problems, then the number of knots is automatically reduced to the closest value without estimation problems.

`logistic.cal` `TRUE` plots the logistic calibration curve, `FALSE` suppresses this curve. Default is `FALSE`.

`lty.log` if `logistic.cal=TRUE`, the linetype of the logistic calibration curve. Default is `1`.

`col.log` if `logistic.cal=TRUE`, the color of the logistic calibration curve. Default is `"black"`.

`lwd.log` if `logistic.cal=TRUE`, the line width of the logistic calibration curve. Default is `1`.

`xlab` x-axis label, default is `"Predicted Probability"`.

`ylab` y-axis label, default is `"Observed proportion"`.

`xlim, ylim` numeric vectors of length 2, giving the x and y coordinates ranges (see `plot.window`)

`m` If grouped proportions are desired, average no. observations per group

`g` If grouped proportions are desired, number of quantile groups

`cuts` If grouped proportions are desired, actual cut points for constructing intervals, e.g. `c(0,.1,.8,.9,1)` or `seq(0,1,by=.2)`

`emax.lim` Vector containing lowest and highest predicted probability over which to compute `Emax`.

`legendloc` if `pl=TRUE`, list with components `x,y` or vector `c(x,y)` for bottom right corner of legend for curves and points. Default is `c(.50, .27)` scaled to lim. Use `locator(1)` to use the mouse, `FALSE` to suppress legend.

`statloc` the "abc" of model performance (Steyerberg et al., 2011)-calibration intercept, calibration slope, and c statistic-will be added to the plot, using statloc as the upper left corner of a box (default is c(0,.85). You can specify a list or a vector. Use locator(1) for the mouse, `FALSE` to suppress statistics. This is plotted after the curve legends.

`dostats` specifies whether and which performance measures are shown in the figure. `TRUE` shows the `"abc"` of model performance (Steyerberg et al., 2011): calibration intercept, calibration slope, and c-statistic. `TRUE` is default. `FALSE` suppresses the presentation of statistics in the figure. A `c()` list of specific stats shows the specified stats. The key stats which are also mentioned in this paper are `"C (ROC)"` for the c statistic, `"Intercept"` for the calibration intercept, `"Slope"` for the calibration slope, and `"ECI"` for the estimated calibration index (Van Hoorde et al, 2015). The full list of possible statistics is taken from `val.prob` and augmented with the estimated calibration index: `"Dxy"`, `"C (ROC)"`, `"R2"`, `"D"`, `"D:Chi-sq"`, `"D:p"`, `"U"`, `"U:Chi-sq"`, `"U:p"`, `"Q"`, `"Brier"`, `"Intercept"`, `"Slope"`, `"Emax"`, `"Brier scaled"`, `"Eavg"`, `"ECI"`. These statistics are always returned by the function.

`cl.level` if `dostats=TRUE`, the confidence level for the calculation of the confidence intervals of the calibration intercept, calibration slope and c-statistic. Default is `0.95`.

`method.ci` method to calculate the confidence interval of the c-statistic. The argument is passed to `auc.nonpara.mw` from the auRoc-package and possible methods to compute the confidence interval are `"newcombe"`, `"pepe"`, `"delong"` or `"jackknife"`. Bootstrap-based methods are not available. The default method is `"pepe"` and here, the confidence interval is the logit-transformation-based confidence interval as documented in Qin and Hotilovac (2008). See `auc.nonpara.mw` for more information on the other methods.

`roundstats` specifies the number of decimals to which the statistics are rounded when shown in the plot. Default is 2.

`riskdist` Use `"calibrated"` to plot the relative frequency distribution of calibrated probabilities after dividing into 101 bins from `lim[1]` to `lim[2]`. Set to `"predicted"` (the default as of rms 4.5-1) to use raw assigned risk, `FALSE` to omit risk distribution. Values are scaled so that highest bar is `0.15*(lim[2]-lim[1])`.

`cex, cex.leg` controls the font size of the statistics (`cex`) or plot legend (`cex.leg`). Default is 0.75

`connect.group` Defaults to `FALSE` to only represent group fractions as triangles. Set to `TRUE` to also connect with a solid line.

`connect.smooth` Defaults to `TRUE` to draw smoothed estimates using a line. Set to `FALSE` to instead use dots at individual estimates

`g.group` number of quantile groups to use when `group` is given and variable is numeric.

`evaluate` number of points at which to store the `lowess`-calibration curve. Default is 100. If there are more than `evaluate` unique predicted probabilities, `evaluate` equally-spaced quantiles of the unique predicted probabilities, with linearly interpolated calibrated values, are retained for plotting (and stored in the object returned by `val.prob`.

`nmin` applies when `group` is given. When `nmin` > 0, `val.prob` will not store coordinates of smoothed calibration curves in the outer tails, where there are fewer than `nmin` raw observations represented in those tails. If for example `nmin`=50, the `plot` function will only plot the estimated calibration curve from a to b, where there are 50 subjects with predicted probabilities $< a$ and $> b$. `nmin` is ignored when computing accuracy statistics.

`d0lab, d1lab` controls the labels for events and non-events (i.e. outcome y) for the histograms. Defaults are `d1lab="1"` for events and `d0lab="0"` for non-events.

`cex.d01` controls the size of the labels for events and non-events. Default is 0.7.

`dist.label` controls the horizontal position of the labels for events and non-events. Default is 0.04.

`line.bins` controls the horizontal (y-axis) position of the histograms. Default is -0.05.

`dist.label2` controls the vertical distance between the labels for events and non-events. Default is 0.03.

`cutoff` puts an arrow at the specified risk cut-off(s). Default is none.

`las` controls whether y-axis values are shown horizontally (1) or vertically (0).

`length.seg` controls the length of the histogram lines. Default is `1`.

`y.intersp` character interspacing for vertical line distances of the legend (`legend`)

`lty.ideal` linetype of the ideal line. Default is `1`.

`col.ideal` controls the color of the ideal line on the plot. Default is `"red"`.

`lwd.ideal` controls the line width of the ideal line on the plot. Default is `1`.

`...` arguments to be passed to `plot`, see `par`

Value

An object of type `CalibrationCurve` with the following slots:

`call` the matched call.

`stats` a vector containing performance measures of calibration.

`cl.level` the confidence level used.

`Calibration` contains the calibration intercept and slope, together with their confidence intervals.

`Cindex` the value of the c-statistic, together with its confidence interval.

Note

In order to make use (of the functions) of the package auRoc, the user needs to install JAGS. However, since our package only uses the `auc.nonpara.mw` function which does not depend on the use of JAGS, we therefore copied the code and slightly adjusted it when `method="pepe"`.

References

Qin, G., & Hotilovac, L. (2008). Comparison of non-parametric confidence intervals for the area under the ROC curve of a continuous-scale diagnostic test. *Statistical Methods in Medical Research*, **17(2)**, pp. 207-21

Steyerberg, E.W., Van Calster, B., Pencina, M.J. (2011). Performance measures for prediction models and markers : evaluation of predictions and classifications. *Revista Espanola de Cardiologia*, **64(9)**, pp. 788-794

Van Calster, B., Nieboer, D., Vergouwe, Y., De Cock, B., Pencina M., Steyerberg E.W. (2016). A calibration hierarchy for risk models was defined: from utopia to empirical data. *Journal of Clinical Epidemiology*, **74**, pp. 167-176

Van Hoorde, K., Van Huffel, S., Timmerman, D., Bourne, T., Van Calster, B. (2015). A spline-based tool to assess and visualize the calibration of multiclass risk predictions. *Journal of Biomedical Informatics*, **54**, pp. 283-93

Examples

```
# Load package
library(CalibrationCurves)
set.seed(1783)

# Simulate training data
X      = replicate(4, rnorm(5e2))
p0true = binomial()$linkinv(cbind(1, X) %*% c(0.1, 0.5, 1.2, -0.75, 0.8))
y      = rbinom(5e2, 1, p0true)
Df     = data.frame(y, X)

# Fit logistic model
FitLog = lrm(y ~ ., Df)

# Simulate validation data
Xval   = replicate(4, rnorm(5e2))
p0true = binomial()$linkinv(cbind(1, Xval) %*% c(0.1, 0.5, 1.2, -0.75, 0.8))
yval   = rbinom(5e2, 1, p0true)
Pred   = binomial()$linkinv(cbind(1, Xval) %*% coef(FitLog))

# Default calibration plot
val.prob.ci.2(Pred, yval)

# Adding logistic calibration curves and other additional features
val.prob.ci.2(Pred, yval, CL.smooth = TRUE, logistic.cal = TRUE, lty.log = 2,
 col.log = "red", lwd.log = 1.5)

val.prob.ci.2(Pred, yval, CL.smooth = TRUE, logistic.cal = TRUE, lty.log = 9,
col.log = "red", lwd.log = 1.5, col.ideal = colors()[10], lwd.ideal = 0.5)
```

Index