Package: reliabilitydiag (via r-universe)

August 22, 2024

Version 0.2.1.9000

Title Reliability Diagrams Using Isotonic Regression

Description Checking the reliability of predictions via the CORP approach, which generates provably statistically 'C'onsistent, 'O'ptimally binned, and 'R'eproducible reliability diagrams using the 'P'ool-adjacent-violators algorithm. See Dimitriadis, Gneiting, Jordan (2021) <doi:10.1073/pnas.2016191118>.

URL https://github.com/aijordan/reliabilitydiag/

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

Depends R (>= 3.5)

Imports magrittr, tidyr, ggplot2, ggExtra, dplyr, purrr, rlang, tibble, vctrs, bde

Suggests monotone

Repository https://aijordan.r-universe.dev

RemoteUrl https://github.com/aijordan/reliabilitydiag

RemoteRef HEAD

RemoteSha a3d35725a5e419e09b2923643d80970655340af5

Contents

`as.reliabilitydiag` *Coerce to a reliability diagram*

Description

Coerce numeric vectors, data frames, or anything else that can be coerced by `as.data.frame` to a data frame of prediction values, into an object inheriting from the `'reliabilitydiag'` class.

Usage

```
as.reliabilitydiag(x, ...)

is.reliabilitydiag(x)

## S3 method for class 'reliabilitydiag'
as.reliabilitydiag(x, y = NULL, r = NULL, tol = sqrt(.Machine$double.eps), ...)

## Default S3 method:
as.reliabilitydiag(
  x,
  y = NULL,
  r = NULL,
  xtype = NULL,
  xvalues = NULL,
  .name_repair = "unique",
  region.level = 0.9,
  region.method = NULL,
  region.position = "diagonal",
  n.boot = 100,
  ...
)

## S3 method for class 'data.frame'
as.reliabilitydiag(
  x,
  y = NULL,
  r = NULL,
  xtype = NULL,
  xvalues = NULL,
  .name_repair = "unique",
  region.level = 0.9,
  region.method = NULL,
  region.position = "diagonal",
  n.boot = 100,
  ...
)
```

Arguments

x	an R object with probability predictions taking values in [0, 1]; usually a numeric vector or a list/data.frame containing numeric vectors.
...	further arguments to be passed to or from methods.
y	a numeric vector of binary response values in {0, 1} to be predicted.
r	an object inheriting from the class 'reliabilitydiag'; alternative to y.
tol	accuracy when comparing y in 'reliabilitydiag' objects.
xtype	a string specifying whether the prediction values should be treated as "continuous" or "discrete".
xvalues	a numeric vector of possible prediction values; values in x are rounded to the nearest value in xvalues and xtype is set to "discrete".
.name_repair	This argument is passed on as repair to vec_as_names. See there for more details.
region.level	a value in (0, 1) specifying the level at which consistency or confidence regions are calculated.
region.method	a string specifying whether "resampling", "continuous_asymptotics", or "discrete_asymptotics" are used to calculate consistency/confidence regions.
region.position	a string specifying whether consistency regions around the "diagonal" or confidence regions around the "estimate" are calculated.
n.boot	the number of bootstrap samples when region.method == "resampling".

Value

as.reliabilitydiag returns a 'reliabilitydiag' object.

is.reliabilitydiag returns TRUE if its argument is a reliability diagram, that is, has "reliabilitydiag" among its classes, and FALSE otherwise.

See Also

reliabilitydiag

---

c.reliabilitydiag *Combining reliability diagram objects*

Description

Combine two or more 'reliabilitydiag' objects that are based on the same observations. Other objects are coerced by as.reliabilitydiag before combination.

Usage

```
## S3 method for class 'reliabilitydiag'
c(
  ...,
  tol = sqrt(.Machine$double.eps),
  xtype = NULL,
  xvalues = NULL,
  region.level = 0.9,
  region.method = NULL,
  region.position = "diagonal",
  n.boot = 100
)
```

Arguments

`...`	objects to be concatenated.
`tol`	accuracy when comparing y in `'reliabilitydiag'` objects.
`xtype`	a string specifying whether the prediction values should be treated as `"continuous"` or `"discrete"`.
`xvalues`	a numeric vector of possible prediction values; values in `x` are rounded to the nearest value in `xvalues` and `xtype` is set to `"discrete"`.
`region.level`	a value in (0, 1) specifying the level at which consistency or confidence regions are calculated.
`region.method`	a string specifying whether `"resampling"`, `"continuous_asymptotics"`, or `"discrete_asymptotics"` are used to calculate consistency/confidence regions.
`region.position`	a string specifying whether consistency regions around the `"diagonal"` or confidence regions around the `"estimate"` are calculated.
`n.boot`	the number of bootstrap samples when `region.method == "resampling"`.

Value

an object inheriting from the class `'reliabilitydiag'`.

See Also

`as.reliabilitydiag`, `[.reliabilitydiag`.

Examples

```
data("precip_Niamey_2016", package = "reliabilitydiag")

X <- precip_Niamey_2016[c("EMOS", "ENS")]
Y <- precip_Niamey_2016$obs
r0 <- reliabilitydiag0(Y)
r1 <- c(r0, X, EPC = precip_Niamey_2016$EPC, region.level = NA)
r1
```

```
c(r1, reliabilitydiag(Logistic = precip_Niamey_2016$Logistic, y = Y))
```

`miscalibration_test` *Miscalibration Test*

Description

(experimental)

Usage

```
miscalibration_test(x, ...)

## S3 method for class 'reliabilitydiag'
miscalibration_test(x, ...)

## S3 method for class 'numeric'
miscalibration_test(x, y, ...)
```

Arguments

`x`	an R object inheriting from `'reliabilitydiag'` or a numeric vector of probability predictions taking values in [0, 1].
`...`	further arguments to be passed to or from methods.
`y`	a numeric vector of binary response values in {0, 1} to be predicted.

Value

returns a `'tibble'` with entries

`forecast`	the name of the prediction method.
`miscalibration`	the miscalibration statistic (see `summary.reliabilitydiag`).
`pvalue`	the pvalue.

`plot.reliabilitydiag` *Plotting reliability diagram objects*

Description

Using the **ggplot2** package to visually diagnose the reliability of prediction methods that issue probability forecasts.

Usage

```
## S3 method for class 'reliabilitydiag'
plot(x, ...)

## S3 method for class 'reliabilitydiag'
autoplot(
  object,
  ...,
  type = c("miscalibration", "discrimination"),
  colour = "red",
  params_histogram = NULL,
  params_ggMarginal = NULL,
  params_ribbon = NULL,
  params_diagonal = NULL,
  params_vsegment = NULL,
  params_hsegment = NULL,
  params_CEPline = NULL,
  params_CEPsegment = NULL,
  params_CEPpoint = NULL
)

## S3 method for class 'reliabilitydiag'
autolayer(
  object,
  ...,
  type = c("miscalibration", "discrimination"),
  colour = "red",
  params_histogram = NA,
  params_ggMarginal = NA,
  params_ribbon = NA,
  params_diagonal = NA,
  params_vsegment = NA,
  params_hsegment = NA,
  params_CEPline = NA,
  params_CEPsegment = NA,
  params_CEPpoint = NA
)
```

Arguments

x	an object inheriting from the class 'reliabilitydiag'.
...	further arguments to be passed to or from methods.
object	an object inheriting from the class 'reliabilitydiag'.
type	one of "miscalibration", "discrimination"; determines which layers are added by default, including default parameter values.
colour	a colour to be used to draw focus; used for the CEP layers when type is "miscalibration", and for the horizontal segment layer and CEP margin histogram when type is "discrimination".
params_histogram	a list of arguments for ggplot2::geom_histogram; this layer shows a histogram of the forecast values in the main plotting region.
params_ggMarginal	a list of arguments for ggExtra::ggMarginal; used to show the marginal distributions of the forecast values and estimated CEP values by adding plots to the top and right of the main plotting region. If this is anything other than NA, the autoplot output cannot be customized by with additional layers.
params_ribbon	a list of arguments for ggplot2::geom_ribbon; this layer shows the uncertainty quantification results.
params_diagonal	a list of arguments for ggplot2::geom_line; this background layer illustrates perfect reliability.
params_vsegment	a list of arguments for ggplot2::geom_segment; this layer shows a vertical segment illustrating the average forecast value.
params_hsegment	a list of arguments for ggplot2::geom_segment; this layer shows a horizontal segment illustrating the average event frequency.
params_CEPline	a list of arguments for ggplot2::geom_line; this layer shows a linear interpolation of the CEP estimates.
params_CEPsegment	a list of arguments for ggplot2::geom_segment; this layer highlights the pieces where the CEP estimate remains constant.
params_CEPpoint	a list of arguments for ggplot2::geom_point; this layer highlights the CEP estimate only for actually observed forecast values.

Details

plot always sends a plot to a graphics device, wheres autoplot behaves as any ggplot() + layer() combination. That means, customized plots should be created using autoplot and autolayer.

Three sets of default parameter values are used:

- If multiple predictions methods are compared, then only the most necessary information to determine reliability are displayed.

- For a single prediction method and `type = "miscalibration"`, the focus lies on the deviation from the diagonal including uncertainty quantification.
- For a single prediction method and `type = "discrimination"`, the focus lies on the PAV transformation and the resulting marginal distribution. A concentration of CEP values near 0 or 1 suggest a high potential predictive ability of a prediction method.

Setting any of the `params_*` arguments to `NA` disables that layer.

Default parameter values if `length(object) > 1`, where the internal variable `forecast` is used as grouping variable:

```
params_histogram    NA
 params_ggMarginal  NA
 params_ribbon      NA
 params_diagonal    list(size = 0.3, colour = "black")
 params_vsegment    NA
 params_hsegment    NA
 params_CEPline     list(size = 0.2)
 params_CEPsegment  NA
 params_CEPpoint    NA
```

Default parameter values for `type = "miscalibration"` if `length(object) == 1`:

```
params_histogram    list(yscale = 0.2, colour = "black", fill = NA)
 params_ggMarginal  NA
 params_ribbon      list(fill = "blue", alpha = 0.15)
 params_diagonal    list(size = 0.3, colour = "black")
 params_vsegment    NA
 params_hsegment    NA
 params_CEPline     list(size = 0.2, colour = colour)
 params_CEPsegment  list(size = 2, colour = colour) if xtype == "continuous"; NA otherwise.
 params_CEPpoint    list(size = 2, colour = colour) if xtype == "discrete"; NA otherwise.
```

Default parameter values for `type = "discrimination"` if `length(object) == 1`:

```
params_histogram    NA
 params_ggMarginal  list(type = "histogram", xparams = list(bins = 100, fill = "grey"), yparams = list(bins =
 params_ribbon      NA
 params_diagonal    list(size = 0.3, colour = "lightgrey")
 params_vsegment    list(size = 1.5, colour = "grey")
 params_hsegment    list(size = 1.5, colour = colour)
 params_CEPline     list(size = 0.2, colour = "black")
 params_CEPsegment  NA
 params_CEPpoint    list(colour = "black")
```

Value

An object inheriting from class `'ggplot'`.

Examples

```
data("precip_Niamey_2016", package = "reliabilitydiag")
r <- reliabilitydiag(
  precip_Niamey_2016[c("Logistic", "EMOS", "ENS", "EPC")],
  y = precip_Niamey_2016$obs,
  region.level = NA
)

# simple plotting
plot(r)

# faceting using the internal grouping variable 'forecast'
autoplot(r, params_histogram = list(colour = "black", fill = NA)) +
  ggplot2::facet_wrap("forecast")

# custom color scale for multiple prediction methods
cols <- c(Logistic = "red", EMOS = "blue", ENS = "darkgreen", EPC = "orange")
autoplot(r) +
  ggplot2::scale_color_manual(values = cols)

# default reliability diagram type with a title
rr <- reliabilitydiag(
  EMOS = precip_Niamey_2016$EMOS,
  r = r,
  region.level = 0.9
)
autoplot(rr) +
  ggplot2::ggtitle("Reliability diagram for EMOS method")

# using defaults for discrimination diagrams
p <- autoplot(r["EMOS"], type = "discrimination")
print(p, newpage = TRUE)

# ggMarginal needs to be called after adding all custom layers
p <- autoplot(r["EMOS"], type = "discrimination", params_ggMarginal = NA) +
  ggplot2::ggtitle("Discrimination diagram for EMOS method")
p <- ggExtra::ggMarginal(p, type = "histogram")
print(p, newpage = TRUE)

# the order of the layers can be changed
autoplot(rr, colour = "black", params_ribbon = NA) +
  autolayer(rr, params_ribbon = list(fill = "red", alpha = .5))
```

`precip_Niamey_2016` *Precipitation forecasts and observations at Niamey, Niger in July to September 2016*

Description

A data set containing 24-hour ahead daily probability of precipitation forecasts of four forecasting methods and corresponding observations of precipitation occurrence.

For a detailed description of the four prediction methods, see Vogel et al (2021).

Usage

```
precip_Niamey_2016
```

Format

A data frame with 92 rows and 6 variables:

`date` a date from `"2016-07-01"` to `"2016-09-30"` in `Date` format.

`Logistic` prediction based on logistic regression, as a probability.

`EMOS` prediction based on EMOS method, as a probability.

`ENS` prediction based on ECMWF raw ensemble, as a probability.

`EPC` prediction based on EPC method, as a probability.

`obs` observation, indicator variable where `1` represents the occurrence of precipitation.

Source

Vogel P, Knippertz P, Gneiting T, Fink AH, Klar M, Schlueter A (2021). "Statistical forecasts for the occurrence of precipitation outperform global models over northern tropical Africa." Geophysical Research Letters, 48, e2020GL091022. doi:10.1029/2020GL091022.

This data set contains modified historic products from the European Center for Medium-Range Weather Forecasts (ECMWF, `https://www.ecmwf.int/`), specifically: ensemble forecasts of precipitation that have been summarized to a probability of precipitation (column `ENS`), and historical observations for the occurence of precipitation (column `obs`). The ECMWF licenses the use of expired real-time data products under the Creative Commons Attribution 4.0 International (CC BY 4.0, `https://creativecommons.org/licenses/by/4.0/`).

`print.reliabilitydiag` *Printing reliability diagram objects*

Description

Printing methods for `'reliabilitydiag'` and `'summary.reliabilitydiag'` objects.

Usage

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

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

Arguments

x	an object inheriting from the class 'reliabilitydiag'.
...	further arguments to be passed to or from methods; in particular, these are passed to autoplot.reliabilitydiag and print.tbl_df.

Details

print.reliabilitydiag always sends a plot to the current graphics device and prints a summary to the console.

print.summary.reliabilitydiag prints the summary output to the console.

Value

Invisibly returns x.

See Also

autoplot.reliabilitydiag, summary.reliabilitydiag

---

reliabilitydiag *Reliability diagram object*

Description

Documentation of the 'reliabilitydiag' object, and its constructors.

Usage

```
reliabilitydiag(
  ...,
  y = NULL,
  r = NULL,
  tol = sqrt(.Machine$double.eps),
  xtype = NULL,
  xvalues = NULL,
  region.level = 0.9,
  region.method = NULL,
  region.position = "diagonal",
  n.boot = 100
)

reliabilitydiag0(y)
```

Arguments

`...`	objects to be coerced to `'reliabilitydiag'` and concatenated
`y`	a numeric vector of binary response values in $\{0, 1\}$ to be predicted.
`r`	an object inheriting from the class `'reliabilitydiag'`; alternative to `y`.
`tol`	accuracy when comparing `y` in `'reliabilitydiag'` objects.
`xtype`	a string specifying whether the prediction values should be treated as `"continuous"` or `"discrete"`.
`xvalues`	a numeric vector of possible prediction values; values in `x` are rounded to the nearest value in `xvalues` and `xtype` is set to `"discrete"`.
`region.level`	a value in (0, 1) specifying the level at which consistency or confidence regions are calculated.
`region.method`	a string specifying whether `"resampling"`, `"continuous_asymptotics"`, or `"discrete_asymptotics"` are used to calculate consistency/confidence regions.
`region.position`	a string specifying whether consistency regions around the `"diagonal"` or confidence regions around the `"estimate"` are calculated.
`n.boot`	the number of bootstrap samples when `region.method == "resampling"`.

Details

`reliabilitydiag` constructs and returns an object inheriting from the class `'reliabilitydiag'`. Each object passed via `...` is coerced by the methods described in `as.reliabilitydiag`, and then concatenated by `c.reliabilitydiag`.

`reliabilitydiag0` constructs an empty `'reliabilitydiag'` object from the response values.

If any of the arguments `region.level`, `region.method`, or `region.position` is `NA`, then the uncertainty quantification in terms of consistency/confidence regions is skipped.

Consistency regions are determined under the assumption of calibration of the original predictions, that is, perfectly reliable forecasts such that $P(Y = 1|X) = X$. Consistency regions are therefore positioned around values on the diagonal (set `region.position` to `"diagonal"`).

For confidence regions, calibration is enforced by using the PAV-recalibrated predictions for uncertainty quantification, that is, it is assumed that $P(Y = 1|X) = PAV(X)$. Confidence regions are therefore positioned around the estimated conditional exceedence probability (CEP) line (set `region.position` to `"estimate"`).

When `region.method` is `"resampling"`, then the original forecast-observations pairs are bootstrapped `n.boot` times. For each bootstrap sample, new observations are drawn under the respective assumption (consistency or confidence). Then PAV-recalibration with those new observations is performed on each bootstrap sample, and pointwise lower and upper bounds are calculated across the resulting CEP lines.

When `region.method` is `"discrete_asymptotics"` and `region.position` is `"diagonal"`, a Gaussian approximation is used assuming $\sqrt{n} * (EST(x) - x)$ has variance $x(1 - x)$, where x is an original prediction value, n is the observed number of predictions with value x, and $EST(x)$ is the estimated CEP value at x.

When `region.method` is `"continuous_asymptotics"` and `region.position` is `"diagonal"`, a Chernoff approximation is used for $(n * f(x)/(4 * x * (1 - x)))^{(1/3)} * (EST(x) - x)$, where x is

an original prediction value, n is the total number of observations, $EST(x)$ is the estimated CEP value at x, and $f(x)$ is the estimated value of the density of the original prediction values. This density is estimated using the bde package: We use Chen's beta kernel density estimator (see bde).

Value

`reliabilitydiag` returns a `'reliabilitydiag'` object, which is a named list-type vector class with the attribute `y` containing the values supplied to the input argument `y`, that is, the numeric vector of response values to be predicted. The length is given by the number of prediction methods detected from the supplied objects.

`reliabilitydiag0` returns an empty `'reliabilitydiag'` object with attribute `y`.

Each entry of a `'reliabilitydiag'` object (corresponding to a single prediction method) is itself a list with the following entries

`cases`	a tibble of all predictions and observations.
`bins`	a tibble of the characteristics of the PAV induced bins.
`regions`	a tibble with lower and upper bounds of the pointwise consistency/confidence regions.
`xinfo`	a list of characteristics of `x`.

Each `cases` tibble comprises the forecast-observation pairs of the given prediction method. It is arranged in increasing order of `x` and has columns

`case_id`	an ID based on the original order of the predictions and observations.
`x`	an original prediction (increasing order).
`y`	an observation, corresponding to `x`.
`bin_id`	an ID for the PAV-recalibration induced bins.
`CEP_pav`	the unique PAV-recalibrated prediction corresponding to `bin_id`.

Each `bins` tibble contains PAV-recalibration information, and has columns

`bin_id`	as in `cases`, with any ID only appearing once.
`n`	the number of predictions with a given `bin_id`.
`x_min`	the smallest value of the predictions with the given `bin_id`.
`x_max`	the largest value of the predictions with the given `bin_id`.
`CEP_pav`	the unique PAV-recalibrated prediction corresponding to `bin_id`.

Each `regions` tibble contains the uncertainty quantification information, and has columns

`x`	an original prediction, with any value only appearing once.
`lower`	the lower bound of the consistency/confidence region at `x`.
`upper`	the upper bound of the consistency/confidence region `x`.
`n`	the number of predictions with a value of `x`.
`level`	the level of the consistency/confidence regions.
`method`	the method used to calculate the consistency/confidence region.
`position`	`"diagonal"` for a consistency region, and `"estimate"` for a confidence region.

Each `xinfo` list has entries

type	the type of predictions, either "discrete" or "continuous".
values	the values supplied to xvalues.

See Also

c.reliabilitydiag, [.reliabilitydiag, plot.reliabilitydiag.

See summary.reliabilitydiag for a decomposition of predictive performance into miscalibration, discrimination, and uncertainty.

Examples

```
data("precip_Niamey_2016", package = "reliabilitydiag")

# standard use with a data.frame
r <- reliabilitydiag(precip_Niamey_2016["EMOS"], y = precip_Niamey_2016$obs)
r

# no consistency/confidence regions
X <- precip_Niamey_2016$EMOS
Y <- precip_Niamey_2016$obs
r1 <- reliabilitydiag(X = X, y = Y, region.level = NA)
r1

# specify predictions via existing reliabilitydiag
r0 <- reliabilitydiag0(Y)
identical(r1, reliabilitydiag(X = X, r = r0, region.level = NA))

# only observation information is used from existing reliabilitydiag
X2 <- precip_Niamey_2016$ENS
r2 <- reliabilitydiag(X2 = X2, r = r, region.level = NA)
r3 <- reliabilitydiag(X2 = X2, r = r0, region.level = NA)
identical(r2, r3)
```

---

summary.reliabilitydiag

Decomposing scores into miscalibration, discrimination and uncertainty

---

Description

An object of class reliabilitydiag contains the observations, the original forecasts, and recalibrated forecasts given by isotonic regression. The function summary.reliabilitydiag calculates quantitative measures of predictive performance, miscalibration, discrimination, and uncertainty, for each of the prediction methods in relation to their recalibrated version.

Usage

```
## S3 method for class 'reliabilitydiag'
summary(object, ..., score = "brier")
```

Arguments

`object`	an object inheriting from the class `'reliabilitydiag'`.
`...`	further arguments to be passed to or from methods.
`score`	currently only "brier" or a vectorized scoring function, that is, `function(observation, prediction)`.

Details

Predictive performance is measured by the mean score of the original forecast values, denoted by S.

Uncertainty, denoted by UNC, is the mean score of a constant prediction at the value of the average observation. It is the highest possible mean score of a calibrated prediction method.

Discrimination, denoted by DSC, is UNC minus the mean score of the PAV-recalibrated forecast values. A small value indicates a low information content (low signal) in the original forecast values.

Miscalibration, denoted by MCB, is S minus the mean score of the PAV-recalibrated forecast values. A high value indicates that predictive performance of the prediction method can be improved by recalibration.

These measures are related by the following equation,

$$S = MCB - DSC + UNC.$$

Score decompositions of this type have been studied extensively, but the optimality of the PAV solution ensures that MCB is nonnegative, regardless of the chosen (admissible) scoring function. This is a unique property achieved by choosing PAV-recalibration.

If deviating from the Brier score as performance metric, make sure to choose a proper scoring rule for binary events, or equivalently, a scoring function with outcome space $\{0, 1\}$ that is consistent for the expectation functional.

Value

A `'summary.reliability'` object, which is also a tibble (see `tibble::tibble()`) with columns:

`forecast`	the name of the prediction method.
`mean_score`	the mean score of the original forecast values.
`miscalibration`	a measure of miscalibration (*how reliable is the prediction method?*), smaller is better.
`discrimination`	a measure of discrimination (*how variable are the recalibrated predictions?*), larger is better.
`uncertainty`	the mean score of a constant prediction at the value of the average observation.

Examples

```
data("precip_Niamey_2016", package = "reliabilitydiag")
r <- reliabilitydiag(
  precip_Niamey_2016[c("Logistic", "EMOS", "ENS", "EPC")],
  y = precip_Niamey_2016$obs,
  region.level = NA
)
```

```
summary(r)
summary(r, score = function(y, x) (x - y)^2)
```

`[.reliabilitydiag` *Subsetting reliability diagram objects*

Description

Subsetting reliability diagram objects

Usage

```
## S3 method for class 'reliabilitydiag'
x[i]
```

Arguments

`x`	an object inheriting from the class `'reliabilitydiag'`.
`i`	index specifying which elements to extract.

Value

an object inheriting from the class `'reliabilitydiag'`.

See Also

`c.reliabilitydiag`.

Examples

```
data("precip_Niamey_2016", package = "reliabilitydiag")

r <- reliabilitydiag(
  precip_Niamey_2016[c("Logistic", "EMOS")],
  y = precip_Niamey_2016$obs
)
length(r)
r[1]
r["EMOS"]
```

Index