

Package ‘CalSim’

February 1, 2019

Type Package

Title The Calibration Simplex

Version 0.2.2

Author Johannes Resin

Maintainer Johannes Resin <johannes.resin@h-its.org>

Depends R (>= 3.3), spatstat

Suggests scoring

Description Generates the calibration simplex (a generalization of the reliability diagram) for three-category probability forecasts, as proposed by Wilks (2013) <doi:10.1175/WAF-D-13-00027.1>.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-02-01 17:33:35 UTC

R topics documented:

calibration_simplex	2
plot.calibration_simplex	3
Index	5

calibration_simplex *Calibration Simplex*

Description

Generates an object of class `calibration_simplex` which can be used to assess the calibration of ternary probability forecasts. The Calibration Simplex can be seen as generalization of the reliability diagram for binary probability forecasts. For details on the interpretation of the calibration simplex cf. Wilks, 2013. Be aware that some minor changes have been made compared to the calibration simplex as suggested by Wilks, 2013 (cf. note below).

Usage

```
calibration_simplex(n, p_a, p_n, p_b, obs, percentagewise)
```

```
## Default S3 method:
```

```
calibration_simplex(n = 10, p_a = NULL, p_n = NULL,  
  p_b = NULL, obs = NULL, percentagewise = FALSE)
```

Arguments

<code>n</code>	A natural number.
<code>p_a</code>	A vector containing the forecasted probabilities for the above-normal (3) category.
<code>p_n</code>	A vector containing the forecasted probabilities for the near-normal (2) category.
<code>p_b</code>	A vector containing the forecasted probabilities for the below-normal (1) category.
<code>obs</code>	A vector containing the observed outcomes (Categories are encoded as 1 (below-normal), 2 (near-normal) and 3 (above-normal)).
<code>percentagewise</code>	Logical, specifying whether probabilities are percentagewise (summing to 100) or not (summing to 1).

Details

Only two of the three forecast probability vectors (`p_a`, `p_b` and `p_n`) need to be specified.

Value

Object of class `calibration_simplex`.

Note

In contrast to the calibration simplex proposed by Daniel S. Wilks, 2013, the simplex has been mirrored at the diagonal through the left bottom hexagon. The miscalibration error is by default calculated precisely (in each bin as the difference of the relative frequencies of each class and the average forecast probabilities) instead of approximately (using Wilks original formula). Approximate errors can be used by setting `true_error = FALSE` when using `plot.calibration_simplex`.

References

Daniel S. Wilks, 2013, The Calibration Simplex: A Generalization of the Reliability Diagram for Three-Category Probability Forecasts, *Weather and Forecasting*, **28**, 1210-1218

See Also

[plot.calibration_simplex](#)

Examples

```
#install.packages("scoring") #if this package is not installed already
data("WeatherProbs", package = "scoring")

#Calibration Simplex for Temperature Forecasts
cst = calibration_simplex(p_a = WeatherProbs$tabv,
                        p_b = WeatherProbs$tblw,
                        obs=WeatherProbs$tcats)

plot(cst)
plot(cst,error_scale=1)

#Calibration Simplex for Precipitation Forecasts
csp = calibration_simplex(n=19,
                        p_a = WeatherProbs$pabv,
                        p_b = WeatherProbs$pbwl,
                        obs=WeatherProbs$pcats)

plot(csp)
plot(csp,error_scale=1)

#Both forecasts are very(!) underconfident. This seems like an unrealistic example!
```

```
plot.calibration_simplex
```

Plot Calibration Simplex

Description

Plot Calibration Simplex

Usage

```
## S3 method for class 'calibration_simplex'
plot(x, true_error = TRUE, error_scale = 0.3,
     min_bin_freq = 10, plot_error_scale = TRUE, scale_area = NULL,
     indicate_bins = TRUE, ...)
```

Arguments

<code>x</code>	Object of class <code>calibration_simplex</code>
<code>true_error</code>	Logical, specifying whether to use true miscalibration errors or approximate miscalibration errors.
<code>error_scale</code>	A number specifying the magnitude of the miscalibration errors (greater 0, usually should be less than 1, cf. note below).
<code>min_bin_freq</code>	A number. Lower bound for (absolute) frequencies, i.e. how many observations have to lie in a bin for it to be plotted.
<code>plot_error_scale</code>	Logical, specifying whether to plot a scale showing the magnitude of miscalibration errors.
<code>scale_area</code>	Optional. A number by which the areas of the points are scaled. Use if points are too small or too big.
<code>indicate_bins</code>	Logical, specifying whether to connect points to their respective bin (center of hexagon).
<code>...</code>	Arguments concerning the title (e.g. <code>main</code> , <code>cex.main</code> , <code>col.main</code> and <code>font.main</code>) and subtitle (e.g. <code>sub</code> , <code>cex.sub</code> , <code>col.sub</code> and <code>font.sub</code>) may be passed here.

Note

For details on the meaning of the error scale, cf. Wilks, 2013, especially Fig. 2. Note that the miscalibration error in each category is in "probability units" (as it is the average difference in relative frequency and forecast probability in each bin).

Index

`calibration_simplex`, [2](#)
`CalSim(calibration_simplex)`, [2](#)
`plot.calibration_simplex`, [2](#), [3](#), [3](#)