Package ‘wbacon’

June 17, 2021

Type   Package
Title  Weighted BACON Algorithms
Version 0.5-1

Description  The BACON algorithms are methods for multivariate outlier
nomination (detection) and robust linear regression by Billor, Hadi,
to weighted problems is due to Beguin and Hulliger (2008)

License GPL (>= 2)

NeedsCompilation yes

URL https://github.com/tobiasschoch/wbacon

BugReports https://github.com/tobiasschoch/wbacon/issues

Encoding UTF-8

Depends R (>= 3.5.0)

Imports stats, graphics, grDevices, hexbin

Suggests modi, robustbase, robustX (>= 1.2-5), cellWise, knitr,
markdown

VignetteBuilder knitr, rmarkdown

Author Tobias Schoch [aut, cre] (<https://orcid.org/0000-0002-1640-3395>),
R-core [cph] (plot.wbaconlm derives from plot.lm)

Maintainer Tobias Schoch <tobias.schoch@gmail.com>

Repository CRAN

Date/Publication 2021-06-17 06:00:02 UTC

R topics documented:

  wbacoon-package .................................................. 2
  is_outlier .......................................................... 2
  median_w ............................................................. 3
  plot.wbaconlm ..................................................... 4
Description

The package **wbacon** implements the BACON algorithms of Billor et al. (2000) and some of the extensions proposed by Béguin and Hulliger (2008).

Details

See **wBACON** to learn more on the BACON method for multivariate outlier nomination (detection).

See **wBACON_reg** to learn more on the BACON method for robust linear regression.

Author(s)

Tobias Schoch

References


Description

Returns a logical vector that indicates which observations were declared outlier by the method.

Usage

```r
is_outlier(object, ...)
## S3 method for class 'wbaconlm'
is_outlier(object, ...)
## S3 method for class 'wbaconmv'
is_outlier(object, ...)
```
**median_w**

Arguments

object  object of class wbaconmv or wbaconlm.
...  additional arguments passed to the method.

Value

A logical vector.

See Also

wBACON_reg and wBACON

Examples

```r
data(swiss)
m <- wBACON(swiss)
is_outlier(m)
```

---

**median_w** \hspace{1cm} **Weighted Median**

Description

median_w computes the weighted population median.

Usage

```r
median_w(x, w, na.rm = FALSE)
```

Arguments

- `x`  [numeric vector] observations.
- `w`  [numeric vector] weights (same length as vector x).
- `na.rm`  [logical] indicating whether NA values should be removed before the computation proceeds (default: FALSE).

Details

Weighted sample median; see quantile_w for more information.

Value

Weighted estimate of the population median.

See Also

quantile_w
**plot.wbaconlm**  

*Plot Diagnostics for an Object of Class wbaconlm*

**Description**

Four plots (selectable by which) are available for an object of class wbaconlm (see wBACON_reg): A plot of residuals against fitted values, a scale-location plot of $\sqrt{\text{[residuals]}}$ against fitted values, a Normal Q-Q plot, and a plot of the standardized residuals versus the robust Mahalanobis distances.

**Usage**

```r
## S3 method for class 'wbaconlm'
plot(x, which = c(1, 2, 3, 4), hex = FALSE,
caption = c("Residuals vs Fitted", "Normal Q-Q", "Scale-Location",
"Standardized Residuals vs Robust Mahalanobis Distance"),
panel = if (add.smooth) function(x, y, ...) 
         panel.smooth(x, y, iter = iter.smooth, ...) else points,
sub.caption = NULL, main = "",
ask = prod(par("mfcol")) < length(which) && dev.interactive(),
...,
id.n = 3, labels.id = names(residuals(x)), cex.id = 0.75,
qqline = TRUE,
add.smooth = getOption("add.smooth"), iter.smooth = 3,
label.pos = c(4, 2), cex.caption = 1, cex.oma.main = 1.25)
```

**Arguments**

- `x` object of class wbaconlm.
- `which` if a subset of the plots is required, specify a subset of the numbers 1:4, [integer].
- `hex` toggle a hexagonally binned plot, [logical], default hex = FALSE.
- `caption` captions to appear above the plots; [character] vector of valid graphics annotations. It can be set to "" or NA to suppress all captions.
- `panel` panel function. The useful alternative to points, panel.smooth can be chosen by add.smooth = TRUE.
- `sub.caption` common title [character]—above the figures if there are more than one; used as sub (s.title) otherwise. If NULL, as by default, a possible abbreviated version of deparse(x$call) is used.
- `main` title to each plot [character]—in addition to caption.
- `ask` [logical]; if TRUE, the user is asked before each plot, see par(ask=.).
- `...` other parameters to be passed through to plotting functions.
- `id.n` number of points to be labelled in each plot, starting with the most extreme, [integer].
- `labels.id` vector of labels [character], from which the labels for extreme points will be chosen. NULL uses observation numbers.
plot.wbaconmv

cex.id magnification of point labels, [numeric].
qline [logical] indicating if a qqline() should be added to the normal Q-Q plot.
add.smooth [logical] indicating if a smoother should be added to most plots; see also panel above.
iter.smooth the number of robustness iterations [integer], the argument iter in panel.smooth()..
lable.pos positioning of labels [numeric], for the left half and right half of the graph respectively, for plots 1-3.
cex.caption controls the size of caption, [numeric].
cex.oma.main controls the size of the sub.caption only if that is above the figures when there is more than one, [numeric].

Details

The plots for which %in% 1:3 are identical with the plot method for linear models (see plot.lm). There you can find details on the implementation and references.

The standardized residuals vs. robust Mahalanobis distance plot (which = 4) has been proposed by Rousseeuw and van Zomeren (1990).

Value

no return value

References


See Also

wBACON_reg

---

plot.wbaconmv Plot Diagnostics for an Object of Class wbaconmv

Description

Two plots (selectable by which) are available for an object of class wbaconmv: (1) Robust distance vs. Index and (2) Robust distance vs. Univariate projection.

Usage

```r
# S3 method for class 'wbaconmv'
plot(x, which = 1:2,
caption = c("Robust distance vs. Index",
  "Robust distance vs. Univariate projection"), hex = FALSE, col = 2,
pch = 19, ask = prod(par("mfcol")) < length(which) && dev.interactive(),
alpha = 0.05, maxiter = 20, tol = 1e-5, ...)
SeparationIndex(object, alpha = 0.05, tol = 1e-5, maxiter = 20)
```
Arguments

- **x**: object of class `wbaconmv`
- **which**: if a subset of the plots is required, specify a subset of the numbers `1:2`, `[integer]`
- **caption**: captions to appear above the plots; `[character]` vector of valid graphics annotations. It can be set to "" or NA to suppress all captions.
- **hex**: toggle the hexagonal bin plot on/off `[logical]` (default: `hex = FALSE`)
- **col**: color of outliers, `[integer]` (default: `col = 2`)
- **pch**: plot character of outliers, code`[integer]` (default: `pch = 19`)
- **ask**: `[logical]`; if TRUE, the user is asked before each plot, see `par(ask=.)`.
- **alpha**: `[numeric]` tuning constant, level of significance, `0 < α < 1`; (default: `alpha = 0.05`).
- **maxiter**: `[integer]` maximal number of iterations (default: `maxiter = 20`).
- **tol**: numerical termination criterion, `[numeric]` (default: `tol = 1e-5`)
- **object**: object of class `wbaconmv`
- **...**: additional arguments passed to the method.

Details

The first plot (which = 1) is a standard diagnostic tool which plots the observations’ index (1:n) against the robust (Mahalanobis) distances; see, e.g., Rousseeuw and van Driessen (1999).

The second plot (which = 2) plots the univariate projection of the data which maximizes the separation criterion for clusters of Qui and Joe (2006) against the robust (Mahalanobis) distances. This plot is due to Willems et al. (2009).

For large data sets, it is recommended to specify the argument `hex = TRUE`. This option shows a hexagonally binned scatterplot in place of the classical scatterplot.

Value

no return value

References


See Also

`wBACON`
predict.wbaconlm

Predicted Values Based on the Weighted BACON Linear Regression

Description

This function does exactly what `predict` does for the linear model `lm`; see `predict.lm` for more details.

Usage

```r
## S3 method for class 'wbaconlm'
predict(object, newdata = NULL, se.fit = FALSE, scale = NULL, df = Inf,
        interval = c("none", "confidence", "prediction"), level = 0.95,
        type = c("response", "terms"), terms = NULL, na.action = na.pass, ...)
```

Arguments

- `object`: Object of class inheriting from "lm"
- `newdata`: An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
- `se.fit`: A switch `[logical]` indicating if standard errors are required.
- `scale`: Scale parameter for std.err. calculation, `[numeric]`.
- `df`: Degrees of freedom for scale, `[integer]`.
- `interval`: Type of interval calculation, `[character]`. Can be abbreviated.
- `level`: Tolerance/confidence level, `[numeric]`.
- `type`: Type of prediction (response or model term), `[character]`. Can be abbreviated.
- `terms`: If `type = "terms"`, which terms (default is all terms), a `[character]` vector.
- `na.action`: function determining what should be done with missing values in `newdata`. The default is to predict `NA`.
- `...`: further arguments passed to `predict.lm`

Value

`predict.wbaconlm` produces a vector of predictions or a matrix of predictions and bounds with column names `fit`, `lwr`, and `upr` if `interval` is set. For `type = "terms"` this is a matrix with a column per term and may have an attribute "constant".

If `se.fit` is `TRUE`, a list with the following components is returned:

- `fit`: vector or matrix as above
- `se.fit`: standard error of predicted means
- `residual.scale`: residual standard deviations
- `df`: degrees of freedom for residual
quantile_w

Weighted Sample Quantiles

Description

quantile_w computes the weighted population quantiles.

Usage

quantile_w(x, w, probs, na.rm = FALSE)

Arguments

x [numeric vector] observations.
w [numeric vector] weights (same length as vector x).
probs [numeric vector] vector of probabilities with values in [0,1].
na.rm [logical] indicating whether NA values should be removed before the computation proceeds (default: FALSE).

Details

Overview. quantile_w computes the weighted sample quantiles; argument probs allows vector inputs.

Implementation. The function is based on a weighted version of the quickselect algorithm with the Bentley and McIlroy (1993) 3-way partitioning scheme. For very small arrays, we use insertion sort.

Compatibility. For equal weighting, i.e. when all elements in w are equal, quantile_w computes quantiles that are identical with type = 2 in stats::quantile; see also Hyndman and Fan (1996).

Value

Weighted estimate of the population quantiles.

See Also

wBACON_reg

Examples

data(iris)
m <- wBACON_reg(Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width, data = iris)
predict(m, newdata = data.frame(Sepal.Width = 1, Petal.Length = 1, Petal.Width = 1))
wBACON

References


See Also

*median_w*

---

**wBACON**

*Weighted BACON Algorithm for Multivariate Outlier Detection*

**Description**

wBACON is an iterative method for the computation of multivariate location and scatter (under the assumption of a Gaussian distribution).

**Usage**

```r
wBACON(x, weights = NULL, alpha = 0.05, collect = 4, version = c("V2", "V1"),
      na.rm = FALSE, maxiter = 50, verbose = FALSE, n_threads = 2)

distance(x)
## S3 method for class 'wbaconmv'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
## S3 method for class 'wbaconmv'
summary(object, ...)
## S3 method for class 'wbaconmv'
center(object)
## S3 method for class 'wbaconmv'
vcov(object, ...)
```

**Arguments**

- **x**  
  [matrix] or [data.frame].
- **weights**  
  [numeric] sampling weight (default weights = NULL).
- **alpha**  
  [numeric] tuning constant, level of significance, 0 < α < 1; (default: alpha = 0.05).
- **collect**  
  determines the size m of the initial subset to be m = collect · p, where p is the number of variables, [integer].
- **version**  
  [character] method of initialization; "V1": weighted Mahalanobis distances (not robust but affine equivariant); "V2" (default): Euclidean norm of the data centered by the coordinate-wise weighted median.
- **na.rm**  
  [logical] indicating whether NA values should be removed before the computation proceeds (default: FALSE).
- **maxiter**  
  [integer] maximal number of iterations (default: maxiter = 50).
verbose  [logical] indicating whether additional information is printed to the console (default: TRUE).

n_threads  [integer] number of threads used for OpenMP (default: 2).

digits  [integer] minimal number of significant digits.

...  additional arguments passed to the method.

object  object of class wbaconmv.

Details

The algorithm is initialized from a set of uncontaminated data. Then the subset is iteratively refined; i.e., additional observations are included into the subset if their Mahalanobis distance is below some threshold (likewise, observations are removed from the subset if their distance larger than the threshold). This process iterates until the set of good data remain stable. Observations not among the good data are outliers; see Billor et al. (2000). The weighted Bacon algorithm is due to Béguin and Hulliger (2008).

The threshold for the (squared) Mahalanobis distances is defined as the standardized chi-square $1 - \alpha$ quantile. All observations whose squared Mahalanobis distances is larger than the threshold are regarded as outliers.

If the sampling weights weights are not explicitly specified (i.e., weights = NULL), they are taken to be 1.0.

Incomplete/missing data: The wBACON cannot deal with missing values. In contrast, function BEM in package modi implements the BACON-EEM algorithm of Béguin and Hulliger (2008), which is tailored to work with outlying and missing values.

If the argument na.rm is set to TRUE the method behaves like na.omit.

Assumptions: The BACON algorithm assumes that the non-outlying data have (roughly) an elliptically contoured distribution (this includes the Gaussian distribution as a special case). "Although the algorithms will often do something reasonable even when these assumptions are violated, it is hard to say what the results mean." (Billor et al., 2000, p. 289)

In line with Billor et al. (2000, p. 290), we use the term outlier "nomination" rather than "detection" to highlight that algorithms should not go beyond nominating observations as potential outliers; see also Béguin and Hulliger (2008). It is left to the analyst to finally label outlying observations as such.

Utility functions and tools: Diagnostic plots are available by the plot method.

The method center and vcov return, respectively, the estimated center/location and covariance matrix.

The distance method returns the robust Mahalanobis distances.

The function is_outlier returns a vector of logicals that flags the nominated outliers.

Value

An object of class wbaconmv with slots

x  see function arguments

weights  see function arguments
center estimated center of the data
dist Mahalanobis distances
n number of observations
p number of variables
alpha see function arguments
subset final subset of outlier-free data
cutoff see function arguments
maxiter number of iterations until convergence
version see functions arguments
collect see functions arguments
cov covariance matrix
converged logical that indicates whether the algorithm converged
call the matched call

References


See Also

plot and is_outlier

Examples

data(swiss)
m <- wBACON(dt)
m
which(is_outlier(m))

---

**wBACON_reg**

*Robust Fitting Linear Regression Models by the BACON Algorithm*

**Description**

The weighted BACON algorithm is a robust method to fit weighted linear regression models. The method is robust against outlier in the response variable and the design matrix (leverage observation).
Usage

wBACON_reg(formula, weights = NULL, data, collect = 4, na.rm = FALSE, alpha = 0.05, version = c("V2", "V1"), maxiter = 50, verbose = FALSE, original = FALSE, n_threads = 2)

## S3 method for class 'wbaconlm'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
## S3 method for class 'wbaconlm'
summary(object, ...)  
## S3 method for class 'wbaconlm'
fitted(object, ...)    
## S3 method for class 'wbaconlm'
residuals(object, ...) 
## S3 method for class 'wbaconlm'
coef(object, ...)      
## S3 method for class 'wbaconlm'
vcov(object, ...)      

Arguments

formula an object of class formula: a symbolic description of the model to be fitted.
weights [numeric] sampling weight (default weights = NULL).
data a data.frame object.
collect determines the size m of the initial subset to be m = collect · p, where p is the number of variables, [integer].
na.rm [logical] indicating whether NA values should be removed before the computation proceeds (default: FALSE).
alpha [numeric] tuning constant, level of significance, 0 < α < 1; (default: alpha = 0.05).
version method to initialize the basic subset, [character]: Version "V1" of Billor et al. (2000) yields affine equivariant but not robust estimators; Version "V1" yields estimators that are robust but not affine equivariant; (default: V2).
maxiter [integer] maximal number of iterations (default: maxiter = 50).
verbose [logical] indicating whether additional information is printed to the console (default: TRUE).
original [logical] if original = TRUE the subset of the m = collect · p smallest observations (small w.r.t. to the Mahalanobis distances) is taken from the subset generated by Algorithm 3 as the basic subset for regression [this is the original method of Billor et al. (2000)]; otherwise (i.e., when original = FALSE) the subset that results from Algorithm 3 of Billor et al. (2000) is taken to be the basic subset for regression (default original = FALSE).
n_threads [integer] number of threads used for OpenMP (default: 2).
digits [integer] minimal number of significant digits.
object object of class wbaconlm.
x object of class wbaconlm.
... additional arguments passed to the method.
Details

First, the wBACON method is applied to the model’s design matrix (having removed the regression intercept/constant, if there is a constant) to establish a subset of observations which is supposed to be free of outliers. Second, the so generated subset is regressed onto the corresponding subset of response variables. The subset is iteratively enlarged to include as many “good” observations as possible.

The original approach of Billor et al. (2000) obtains by specifying the argument original = TRUE.

Models for wBACON_reg are specified symbolically. A typical model has the form response ~ terms, where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response.

A formula has an implied intercept term. To remove this use either y ~ x -1 or y ~ 0 + x. See formula or lm for for more details.

The weights argument can be used to specify sampling weights or case weights.

It is not possible to fit multiple response variables (on the r.h.s. of the formula, i.e. multivariate models) in one call.

The method cannot deal with missing values. If the argument na.rm is set to TRUE the method behaves like na.omit.

Assumptions: The algorithm assumes that the non-outlying data follow a linear (homoscedastic) regression model and that the independent variables have (roughly) an elliptically contoured distribution. “Although the algorithms will often do something reasonable even when these assumptions are violated, it is hard to say what the results mean.” (Billor et al., 2000, p. 289)

In line with Billor et al. (2000, p. 290), we use the term outlier “nomination” rather than “detection” to highlight that algorithms should not go beyond nominating observations as potential outliers. It is left to the analyst to finally label outlying observations as such.

Utility functions and tools: The generic functions coef, fitted, residuals, and vcov extract the estimate coefficients, fitted values, residuals, and the covariance matrix of the estimated coefficients.

The function summary summarizes the estimated model.

Value

An object of class wbaconlm with slots

- coefficients: a named vector of coefficients
- residuals: the residuals (for all observations in the data.frame not only the ones in the final subset
- rank: the numeric rank of the fitted linear model (i.e. number of variables in the design matrix
- fitted.values: fitted values
- df.residual: the residual degrees of freedom (computed for the observations in the final subset
- call: the matched call
- terms: the terms object
model  
weights
qr
subset
reg
mv

the `model.frame` used
weights
the `qr` object of the linear model fit for the final subset
the subset
a list with additional details on `wBACON_reg`
a list with details on the results of `wBACON` that have been used to initialize `wBACON_reg`

References


See Also

`plot` gives diagnostic plots for an `wbaconlm` object.

`predict` is used for prediction (incl. confidence and prediction intervals).

Examples

data(iris)
m <- wBACON_reg(Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width, data = iris)
m
summary(m)
Index

BEM, 10

center (wBACON), 9
coeff.wbaconlm (wBACON_reg), 11
distance (wBACON), 9
fitted.wbaconlm (wBACON_reg), 11
formula, 13
is_outlier, 2, 10, 11
lm, 13

median_w, 3, 9
model.frame, 14

na.omit, 10, 13

panel.smooth, 4, 5
par, 4, 6
plot, 10, 11, 14
plot.lm, 5
plot.wbaconlm, 4
plot.wbaconmv, 5
points, 4
predict, 7, 14
predict.lm, 7
predict.wbaconlm, 7
print.wbaconlm (wBACON_reg), 11
print.wbaconmv (wBACON), 9

qqline, 5
qr, 14
quantile_w, 3, 8

residuals.wbaconlm (wBACON_reg), 11

SeparationIndex (plot.wbaconmv), 5
summary.wbaconlm (wBACON_reg), 11
summary.wbaconmv (wBACON), 9

terms, 13
title, 4

vcov.wbaconlm (wBACON_reg), 11
vcov.wbaconmv (wBACON), 9

wBACON, 2, 3, 6, 9, 13, 14
wbacon (wbacon-package), 2
wbacon-package, 2
wBACON_reg, 2–5, 8, 11

wBACON, 2, 3, 6, 9, 13, 14
wbacon (wbacon-package), 2
wbacon-package, 2
wBACON_reg, 2–5, 8, 11

15