Package: wbacon (via r-universe)

October 29, 2026

Type Package

Title Weighted BACON Algorithms

Version 0.6-3

Description The BACON algorithms are methods for multivariate outlier nomination (detection) and robust linear regression by Billor, Hadi, and Velleman (2000) <doi:10.1016/S0167-9473(99)00101-2>. The extension to weighted problems is due to Beguin and Hulliger (2008) <https://www150.statcan.gc.ca/n1/en/catalogue/12-001-X200800110616>; see also <doi:10.21105/joss.03238>.

License GPL (>= 2)

NeedsCompilation yes

LazyData true

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), knitr, rmarkdown

VignetteBuilder knitr, rmarkdown

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

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

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

Date/Publication 2025-05-03 21:30:01 UTC

RemoteUrl https://github.com/cran/wbacon

RemoteRef HEAD

RemoteSha 850d13d7747a50448925ab6b99af78a2cbc4a52f

Contents

wbacon-package . . . 2
is_outlier . . . 3
median_w . . . 4
philips . . . 4
plot.wbaconlm . . . 5
plot.wbaconmv . . . 7
predict.wbaconlm . . . 9
quantile_w . . . 10
wBACON . . . 11
wBACON_reg . . . 14

Index **17**

`wbacon-package` *Weighted BACON Algorithms for Multivariate Outlier Nomination (Detection) and Robust Linear Regression*

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

Billor N., Hadi A.S. and Vellemann P.F. (2000). BACON: Blocked Adaptive Computationally efficient Outlier Nominators. *Computational Statistics and Data Analysis*, **34**, pp. 279–298. doi:10.1016/S01679473(99)001012

Béguin C. and Hulliger B. (2008). The BACON-EEM Algorithm for Multivariate Outlier Detection in Incomplete Survey Data. *Survey Methodology*, **34**, pp. 91–103. https://www150.statcan.gc.ca/n1/en/catalogue/12-001-X200800110616

Schoch, T. (2021). wbacon: Weighted BACON algorithms for multivariate outlier nomination (detection) and robust linear regression, *Journal of Open Source Software*, **6** (62), 3238 doi:10.21105/joss.03238

`is_outlier` *Flag Outliers*

Description

By default the function returns a logical vector that indicates which observations were identified or declared as (potential) outliers by the method; if `names = TRUE` is set in the function call, the row names of the (potential) outliers are returned.

Usage

```
is_outlier(object, ...)
## S3 method for class 'wbaconmv'
is_outlier(object, names = FALSE, ...)
## S3 method for class 'wbaconlm'
is_outlier(object, names = FALSE, ...)
```

Arguments

`object`	object of class `wbaconmv` or `wbaconlm`.
`names`	`[logical]`, if `TRUE` (default), a vector is returned that identifies the (potential) outliers; otherwise, the row names of the outliers are returned.
`...`	additional arguments passed to the method.

Value

A logical vector or vector with row names.

See Also

`wBACON_reg` and `wBACON`

Examples

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

# indicator vector of potential outliers
is_outlier(m)

# names of the potential outliers
is_outlier(m, names = TRUE)
```

median_w *Weighted Median*

Description

median_w computes the weighted population median.

Usage

```
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

philips *Philips data*

Description

The data set consists of 677 observations on 9 variables/characteristics of diaphragm parts for television sets.

Usage

```
data(philips)
```

Format

A `data.frame` with 677 observations on the following variables:

`X1` [double], characteristic 1.

`X2` [double], characteristic 2.

`X3` [double], characteristic 3.

`X4` [double], characteristic 4.

`X5` [double], characteristic 5.

`X6` [double], characteristic 6.

`X7` [double], characteristic 7.

`X8` [double], characteristic 8.

`X9` [double], characteristic 9.

Details

The data have been studied in Rousseeuw and van Driessen (1999) and Billor et al. (2000). They have been published in Raymaekers and Rousseeuw (2023).

Source

Billor, N., A. S. Hadi, and P. F. Vellemann (2000). BACON: Blocked Adaptive Computationally-efficient Outlier Nominators. *Computational Statistics and Data Analysis*, **34**, 279–298. doi:10.1016/S01679473(99)001012

Raymaekers, J. and P. Rousseeuw (2023). cellWise: Analyzing Data with Cellwise Outliers. R package version 2.5.3, https://CRAN.R-project.org/package=cellWise

Rousseeuw, P. J. and K. van Driessen (1999). A fast algorithm for the Minimum Covariance Determinant estimator. *Technometrics*, **41**, 212–223. doi:10.2307/1270566

Examples

```
head(philips)
```

`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 $\sqrt{|residuals|}$ against fitted values, a Normal Q-Q plot, and a plot of the standardized residuals versus the robust Mahalanobis distances.

Usage

```
## 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	toogle 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 *ask*ed 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.
cex.id	magnification of point labels, [numeric].
qqline	[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().
label.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

Rousseeuw, P.J. and B.C. van Zomeren (1990). Unmasking Multivariate Outliers and Leverage Points, *Journal of the American Statistical Association*, **411**, 633–639. doi:10.2307/2289995

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

```
## 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`	toogle the hexagonal bin plot on/off `[logical]` (default: `hex = FALSE`)
`col`	color of outliers, `[integer]` (default: `col = 2`)

pch	plot character of outliers, [integer] (default: pch = 19)
ask	[logical]; if TRUE, the user is *ask*ed before each plot, see par(ask=.).
alpha	[numeric] tuning constant, level of significance, $0 < \alpha < 1$; (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

Rousseeuw, P.J. and K. van Driessen (1999). A Fast Algorithm for the Minimum Covariance Determinant, *Technometrics*, **41**, 212–223. doi:10.2307/1270566

Qiu, W. and H. Joe (2006). Separation index and partial membership for clustering, *Computational Statistics and Data Analysis*, **50**, 585–603. doi:10.1016/j.csda.2004.09.009

Willems, G., H. Joe, and R. Zamar (2009). Diagnosing Multivariate Outliers Detected by Robust Estimators, *Journal of Computational and Graphical Statistics*, **18**, 73–91. doi:10.1198/jcgs.2009.0005

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

```
## S3 method for class 'wbaconlm'
predict(object, newdata, 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

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))
```

---

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.

References

Bentley, J.L. and D.M. McIlroy (1993). Engineering a Sort Function, *Software - Practice and Experience*, **23**, 1249–1265. doi:10.1002/spe.4380231105

Hyndman, R.J. and Y. Fan (1996). Sample Quantiles in Statistical Packages, *The American Statistician*, **50**, 361–365.doi:10.2307/2684934

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

```
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, ...)
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, $0 < \alpha < 1$; (default: alpha = 0.05).
collect	determines the size m of the initial subset to be $m = collect \cdot p$, 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

Billor N., Hadi A.S. and Vellemann P.F. (2000). BACON: Blocked Adaptive Computationally efficient Outlier Nominators. *Computational Statistics and Data Analysis*, **34**, pp. 279–298. doi:10.1016/S01679473(99)001012

Béguin C. and Hulliger B. (2008). The BACON-EEM Algorithm for Multivariate Outlier Detection in Incomplete Survey Data. *Survey Methodology*, **34**, pp. 91–103. https://www150.statcan.gc.ca/n1/en/catalogue/12-001-X200800110616

Schoch, T. (2021). wbacon: Weighted BACON algorithms for multivariate outlier nomination (detection) and robust linear regression, *Journal of Open Source Software*, **6** (62), 3238 doi:10.21105/joss.03238

See Also

plot and is_outlier

Examples

```
data(swiss)
dt <- swiss[, c("Fertility", "Agriculture", "Examination", "Education",
    "Infant.Mortality")]
m <- wBACON(dt)
m

# indicator vector of potential outliers
is_outlier(m)

# names of the potential outliers
is_outlier(m, names = TRUE)
```

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 $m = collect \cdot p$, 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, $0 < \alpha < 1$; (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 $m = collect \cdot p$ 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`	the `model.frame` used
`weights`	weights
`qr`	the `qr` object of the linear model fit for the final subset
`subset`	the subset
`reg`	a list with additional details on `wBACON_reg`
`mv`	a list with details on the results of `wBACON` that have been used to initialize `wBACON_reg`

References

Billor N., Hadi A.S. and Vellemann P.F. (2000). BACON: Blocked Adaptive Computationally efficient Outlier Nominators. *Computational Statistics and Data Analysis*, **34**, pp. 279–298. doi:10.1016/S01679473(99)001012

Schoch, T. (2021). wbacon: Weighted BACON algorithms for multivariate outlier nomination (detection) and robust linear regression, *Journal of Open Source Software*, **6** (62), 3238 doi:10.21105/joss.03238

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

# model summary
summary(m)

# names of potential outliers
is_outlier(m, names = TRUE)
```

Index

∗ **datasets**
philips, 4

BEM, *12*

center *(*wBACON*)*, 11
coef.wbaconlm *(*wBACON_reg*)*, 14

distance *(*wBACON*)*, 11

fitted.wbaconlm *(*wBACON_reg*)*, 14
formula, *15*

is_outlier, 3, *12, 13*

lm, *15*

median_w, 4, *11*
model.frame, *16*

na.omit, *12*, *15*

panel.smooth, *6*
par, *6*, *8*
philips, 4
plot, *12, 13*, *16*
plot.lm, *7*
plot.wbaconlm, 5
plot.wbaconmv, 7
points, *6*
predict, *9*, *16*
predict.lm, *9*
predict.wbaconlm, 9
print.wbaconlm *(*wBACON_reg*)*, 14
print.wbaconmv *(*wBACON*)*, 11

qqline, *6*
qr, *16*
quantile_w, *4*, 10

residuals.wbaconlm *(*wBACON_reg*)*, 14

SeparationIndex *(*plot.wbaconmv*)*, 7
summary.wbaconlm *(*wBACON_reg*)*, 14
summary.wbaconmv *(*wBACON*)*, 11

terms, *16*
title, *6*

vcov.wbaconlm *(*wBACON_reg*)*, 14
vcov.wbaconmv *(*wBACON*)*, 11

wBACON, *2, 3*, *8*, 11, *15, 16*
wbacon *(*wbacon-package*)*, 2
wbacon-package, 2
wBACON_reg, *2, 3*, *5*, *7*, *10*, 14