

Package ‘ib’

September 4, 2025

Type Package

Title Bias Correction via Iterative Bootstrap

Version 0.2.1

Description An implementation of the iterative bootstrap procedure of Kuk (1995) <[doi:10.1111/j.2517-6161.1995.tb02035.x](https://doi.org/10.1111/j.2517-6161.1995.tb02035.x)> to correct the estimation bias of a fitted model object. This procedure has better bias correction properties than the bootstrap bias correction technique.

Maintainer Samuel Orso <Samuel.Orso@unige.ch>

Depends R (>= 4.0.0)

Imports betareg, lme4, Formula, MASS, Matrix, methods, Rdpack (>= 0.7), stats, utils, VGAM

Suggests testthat (>= 3.0.0), knitr, nlraa, rmarkdown

License GPL (>= 2)

Encoding UTF-8

RdMacros Rdpack

RoxygenNote 7.3.2

URL <https://smac-group.github.io/ib/>,
<https://github.com/SMAC-Group/ib/>

BugReports <https://github.com/SMAC-Group/ib/issues/>

Config/testthat/edition 3

NeedsCompilation no

Author Samuel Orso [aut, cre],
Stéphane Guerrier [ctb],
Yuming Zhang [ctb]

Repository CRAN

Date/Publication 2025-09-04 14:00:02 UTC

Contents

bootstrap	3
coef,Ib-method	3
effects,Ib-method	4
fitted,Ib-method	4
getEst	5
getExtra	5
getIteration	6
getObject	7
ib	7
ib,negbin-method	12
Ib-class	12
ibControl	13
plot,Ib,ANY-method	15
predict,Ib-method	15
residuals,Ib-method	16
show,Ib-method	16
show,SummaryIb-method	17
simulation	17
simulation,betareg-method	18
simulation,glm-method	18
simulation,glmerMod-method	19
simulation,lm-method	19
simulation,lmerMod-method	20
simulation,negbin-method	20
simulation,nls-method	21
simulation,vglm-method	21
summary,IbBetareg-method	22
summary,IbGlm-method	22
summary,IbGlmer-method	23
summary,IbLm-method	23
summary,IbLmer-method	24
summary,IbNegbin-method	24
summary,IbNls-method	25
summary,IbVglm-method	25
SummaryIb-class	26
vcov,Ib-method	26

Index

27

bootstrap	<i>Parametric bootstrap</i>
-----------	-----------------------------

Description

Method for generating parametric bootstrap estimates from a fitted model.

Usage

```
bootstrap(object, B = 1000, extra_param = FALSE, ...)
```

Arguments

object	an object representing a fitted model (see 'Details').
B	an integer for number of bootstrap replicates (default 1,000).
extra_param	if TRUE, bootstrap is also performed for extra parameters (see 'Details').
...	additional optional arguments to pass to <code>ibControl</code> .

Details

This method is a simple wrapper around the `ib` method where number of iterations is set to 1.

Value

A matrix `p` (size of parameter) times `B` of bootstrapped estimates.

Author(s)

Samuel Orso

See Also

[ib](#), [ibControl](#)

coef, Ib-method	<i>Method for extracting coefficients from an object in class union "Ib"</i>
-----------------	--

Description

Method for extracting coefficients from an object in class union "Ib"

Usage

```
## S4 method for signature 'Ib'
coef(object, ...)
```

Arguments

object	an object of class union "Ib"
...	further arguments to pass to coef

See Also

[Ib, coef](#)

effects,Ib-method	<i>Method for extracting effects from an object in class union "Ib"</i>
-------------------	---

Description

Method for extracting effects from an object in class union "Ib"

Usage

```
## S4 method for signature 'Ib'
effects(object, ...)
```

Arguments

object	an object of class union "Ib"
...	further arguments to pass to effects

See Also

[Ib, effects](#)

fitted,Ib-method	<i>Method for extracting fitted values from an object in class union "Ib"</i>
------------------	---

Description

Method for extracting fitted values from an object in class union "Ib"

Usage

```
## S4 method for signature 'Ib'
fitted(object, ...)
```

Arguments

object	an object of class union "Ib"
...	further arguments to pass to fitted

See Also

[Ib](#), [fitted.values](#)

getEst

Accessor to the object in class union "Ib"

Description

Method for obtaining estimates from fitted model within any object of class union [Ib](#).

Usage

```
getEst(x)

## S4 method for signature 'Ib'
getEst(x)
```

Arguments

x an object of class union "Ib"

Details

This methods allow to access extra parameter estimates. If extra_param=TRUE, it becomes equivalent to [coef](#).

Value

an estimate (as in [getExtra](#)).

See Also

[Ib](#)

getExtra

Accessor to an extra part in class union "Ib"

Description

Method for obtaining a extra values generated by the iterative bootstrap procedure within any object of class union [Ib](#).

Usage

```
getExtra(x)

## S4 method for signature 'Ib'
getExtra(x)
```

Arguments

x an object of class union "Ib"

Value

a list with the following components:

iteration	number of iterations (k)
of	value of the objective function $ \hat{\pi} - \frac{1}{H} \sum_{h=1}^H \hat{\pi}_h(\hat{\theta}^k) $
estimate	value of the estimates $\hat{\theta}^k$
test_theta	value for difference of thetas: $ \hat{\theta}^k - \hat{\theta}^{k-1} $
ib_warn	optional warning message
boot	matrix of H bootstrap estimates: $\hat{\pi}(\hat{\theta}^k)$

See Also

[Ib](#)

getIteration	<i>Accessor to the object in class union "Ib"</i>
--------------	---

Description

Method for obtaining the number of iteration from fitted model within any object of class union [Ib](#).

Usage

```
getIteration(x)

## S4 method for signature 'Ib'
getIteration(x)
```

Arguments

x an object of class union "Ib"

Details

This methods allow to access extra information about the number of iterations.

Value

a number of iterations (as in [getExtra](#)).

See Also

[Ib](#)

getObject	<i>Accessor to the object in class union "Ib"</i>
-----------	---

Description

Method for obtaining a fitted model within any object of class union [Ib](#).

Usage

```
getObject(x)

## S4 method for signature 'Ib'
getObject(x)
```

Arguments

x an object of class union "Ib"

See Also

[Ib](#)

ib	<i>Bias correction via iterative bootstrap</i>
----	--

Description

ib is used to correct the bias of a fitted model object with the iterative bootstrap procedure.

Usage

```

ib(object, thetastart = NULL, control = list(...), extra_param = FALSE, ...)

## S4 method for signature 'betareg'
ib(object, thetastart = NULL, control = list(...), extra_param = FALSE, ...)

## S4 method for signature 'glm'
ib(object, thetastart = NULL, control = list(...), extra_param = FALSE, ...)

## S4 method for signature 'glmerMod'
ib(object, thetastart = NULL, control = list(...), extra_param = FALSE, ...)

## S4 method for signature 'lm'
ib(object, thetastart = NULL, control = list(...), extra_param = FALSE, ...)

## S4 method for signature 'lmerMod'
ib(object, thetastart = NULL, control = list(...), extra_param = FALSE, ...)

## S4 method for signature 'nls'
ib(object, thetastart = NULL, control = list(...), extra_param = FALSE, ...)

## S4 method for signature 'vglm'
ib(object, thetastart = NULL, control = list(...), extra_param = FALSE, ...)

```

Arguments

<code>object</code>	an object representing a fitted model (see 'Details').
<code>thetastart</code>	an optional starting value for the iterative procedure. If <code>NULL</code> (default), the procedure starts at the estimates in <code>object</code> .
<code>control</code>	a list of parameters for controlling the iterative procedure (see ibControl).
<code>extra_param</code>	if <code>TRUE</code> , the bias of estimation of extra parameters is performed (see 'Details').
<code>...</code>	additional optional arguments (see 'Details').

Details

The iterative bootstrap procedure is described in Kuk (1995) and further studied by Guerrier et al. (2019) and Guerrier et al. (2020). The k th iteration of this algorithm is

$$\hat{\theta}^k = \hat{\theta}^{k-1} + \hat{\pi} - \frac{1}{H} \sum_{h=1}^H \hat{\pi}_h(\hat{\theta}^{k-1})$$

for $k = 1, 2, \dots$ and where the sum is over $h = 1, \dots, H$. The estimate $\hat{\pi}$ is provided by the object. The value $\hat{\pi}_h(\hat{\theta})$ is a parametric bootstrap estimate where the bootstrap sample is generated from $\hat{\theta}$ and a fixed seed (see [ibControl](#)). The greater the parameter value H generally the better bias correction but the more computation it requires (see [ibControl](#)). If `thetastart=NULL`, the initial value of the procedure is $\hat{\theta}^0 = \hat{\pi}$. The number of iterations are controlled by `maxit` parameter of [ibControl](#).

By default, the method correct `coefficients` only. For extra parameters, it depends on the model. These extra parameters may have some constraints (e.g. positivity). If `constraint=TRUE` (see `ibControl`), then a transformation from the constraint space to the real is used for the update.

For `betareg`, `extra_param` is not available as by default mean and precision parameters are corrected. Currently the 'identity' link function is not supported for precision parameters.

For `glm`, if `extra_param=TRUE`: the shape parameter for the `Gamma`, the variance of the residuals in `lm` or the overdispersion parameter of the negative binomial regression in `glm.nb`, are also corrected. Note that the `quasi` families are not supported for the moment as they have no simulation method (see `simulate`). Bias correction for extra parameters of the `inverse.gaussian` is not yet implemented.

For `glmer`, by default, only the fixed effects are corrected. If `extra_param=TRUE`: all the random effects (variances and correlations) are also corrected.

For `lm`, if `extra_param=TRUE`: the variance of the residuals is also corrected. Note that using the `ib` is not useful as coefficients are already unbiased, unless one considers different data generating mechanism such as censoring, missing values and outliers (see `ibControl`).

For `lmer`, by default, only the fixed effects are corrected. If `extra_param=TRUE`: all the random effects (variances and correlations) and the variance of the residuals are also corrected. Note that using the `ib` is certainly not useful with the argument `REML=TRUE` in `lmer` as the bias of variance components is already addressed, unless one considers different data generating mechanism such as censoring, missing values and outliers (see `ibControl`).

For `nls`, if `extra_param=TRUE`: the variance of the residuals is also corrected.

For `vglm`, `extra_param` is currently not used. Indeed, the philosophy of a vector generalized linear model is to potentially model all parameters of a distribution with a linear predictor. Hence, what would be considered as an extra parameter in `glm` for instance, may already be captured by the default coefficients. However, correcting the bias of a coefficients does not imply that the bias of the parameter of the distribution is corrected (by **Jensen's inequality**), so we may use this feature in a future version of the package. Note that we currently only support distributions with a `simslot` (see `simulate.vlm`).

Value

A fitted model object of class `lb`.

Author(s)

Samuel Orso

References

Guerrier S, Dupuis-Lozeron E, Ma Y, Victoria-Feser M (2019). "Simulation-Based Bias Correction Methods for Complex Models." *Journal of the American Statistical Association*, **114**(525), 146-157. doi:10.1080/01621459.2017.1380031, <https://doi.org/10.1080/01621459.2017.1380031>.

Guerrier S, Karemera M, Orso S, Victoria-Feser M, Zhang Y (2020). "A General Approach for Simulation-based Bias Correction in High Dimensional Settings." <https://arxiv.org/pdf/2010.13687.pdf>. Version 2: 13 Nov 2020, 2010.13687, <https://arxiv.org/pdf/2010.13687.pdf>.

Kuk AYC (1995). “Asymptotically Unbiased Estimation in Generalized Linear Models with Random Effects.” *Journal of the Royal Statistical Society: Series B (Methodological)*, **57**(2), 395-407. doi:10.1111/j.25176161.1995.tb02035.x, <https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1995.tb02035.x>, <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1995.tb02035.x>.

See Also

betareg
glm, glm.nb
glmer
lm
lmer
nls
vglm

Examples

```
## beta regression
library(betareg)
data("GasolineYield", package = "betareg")
## currently link.phi = "identity" is not supported
## fit_beta <- betareg(yield ~ batch + temp, data = GasolineYield)
fit_beta <- betareg(yield ~ batch + temp, link.phi = "log", data = GasolineYield)
fit_ib <- ib(fit_beta)

# precision parameter can also depend on covariates
fit_beta <- betareg(yield ~ batch + temp | temp, data = GasolineYield)
fit_ib <- ib(fit_beta)
## poisson regression
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
pois_fit <- glm(counts ~ outcome + treatment, family = poisson())
fit_ib <- ib(pois_fit)
summary(fit_ib)
## Set H = 1000
## Not run:
fit_ib <- ib(pois_fit, control=list(H=1000))
summary(fit_ib)

## End(Not run)

## gamma regression
clotting <- data.frame(
  u = c(5,10,15,20,30,40,60,80,100),
  lot1 = c(118,58,42,35,27,25,21,19,18),
  lot2 = c(69,35,26,21,18,16,13,12,12))
fit_gamma <- glm(lot2 ~ log(u), data = clotting, family = Gamma(link = "inverse"))
fit_ib <- ib(fit_gamma)
```

```

## summary(fit_ib)
## correct for shape parameter and show iterations
## Not run:
fit_ib <- ib(fit_gamma, control=list(verbose=TRUE), extra_param = TRUE)
summary(fit_ib)

## End(Not run)

## negative binomial regression
library(MASS)
fit_nb <- glm.nb(Days ~ Sex/(Age + Eth*Lrn), data = quine)
fit_ib <- ib(fit_nb)
## summary(fit_ib)
## correct for overdispersion with H=100
## Not run:
fit_ib <- ib(fit_nb, control=list(H=100), extra_param = TRUE)
summary(fit_ib)

## End(Not run)

## generalized linear mixed-effects regression
## Not run:
library(lme4)
fit_glmm <- glmer(incidence / size ~ period + (1 | herd), weights = size,
                  family = binomial, data = cbpp)
fit_ib <- ib(fit_glmm)
summary(fit_ib)
## correct for variances and correlation
fit_ib <- ib(fit_glmm, extra_param = TRUE)
summary(fit_ib)

## End(Not run)

## linear regression
fit_lm <- lm(displacement ~ cyl + hp + wt, data = mtcars)
fit_ib <- ib(fit_lm)
summary(fit_ib)
## correct for variance of residuals
fit_ib <- ib(fit_lm, extra_param = TRUE)
summary(fit_ib)

## linear mixed-effects regression
## Not run:
library(lme4)
fit_lmm <- lmer(Reaction ~ Days + (Days | Subject), data = sleepstudy, REML = FALSE)
fit_ib <- ib(fit_lmm)
summary(fit_ib)
## correct for variances and correlation
fit_ib <- ib(fit_lmm, extra_param = TRUE)
summary(fit_ib)

## End(Not run)

```

```
## nonlinear regression
DNase1 <- subset(DNase, Run == 1)
fit_nls <- nls(density ~ SSlogis(log(conc), Asym, xmid, scal), data = DNase1)
fit_ib <- ib(fit_nls)
summary(fit_ib)

## student regression
library(VGAM)
tdata <- data.frame(x = runif(nn <- 1000))
tdata <- transform(tdata,
  y = rt(nn, df = exp(exp(0.5 - x))))
fit_vglm <- vglm(y ~ x, studentt3, data = tdata)
fit_ib <- ib(fit_vglm)
summary(fit_ib)
```

ib,negbin-method	ib method for negbin object from glm.nb function of MASS package.
------------------	--

Description

[ib](#) method for negbin object from [glm.nb](#) function of **MASS** package.

Usage

```
## S4 method for signature 'negbin'
ib(object, thetastart = NULL, control = list(...), extra_param = FALSE, ...)
```

Arguments

object	an object representing a fitted model (see 'Details').
thetastart	an optional starting value for the iterative procedure. If NULL (default), the procedure starts at the estimates in object.
control	a list of parameters for controlling the iterative procedure (see ibControl).
extra_param	if TRUE, the bias of estimation of extra parameters is performed (see 'Details').
...	additional optional arguments (see 'Details').

Ib-class	<i>Fitted-model wrappers for the iterative bootstrap (ib)</i>
----------	---

Description

S4 classes that wrap fitted model objects and attach extra metadata from the iterative bootstrap procedure.

Members: [IbBetareg](#), [IbGlm](#), [IbLm](#), [IbLmer](#), [IbGlmer](#), [IbNegbin](#), [IbNls](#), [IbVglm](#).

Details

Members include wrappers around models from **betareg**, **stats**, **lme4**, **MASS**, and **VGAM**.

These classes are light wrappers: they store the corrected fitted object in @object (see [getObject](#)) and additional information in @ib_extra (see [getExtra](#)).

Value

Objects of the respective S4 classes.

Slots

object The original fitted model object, corrected by ib.

ib_extra A list of extra metadata from ib.

See Also

[getObject](#), [getExtra](#), [Ib](#), [SummaryIb](#)

 ibControl

Auxiliary for controlling IB

Description

Auxiliary function for [ib](#) bias correction.

Usage

```
ibControl(
  tol = 1e-05,
  maxit = 25,
  verbose = FALSE,
  seed = 123L,
  H = 1L,
  constraint = TRUE,
  early_stop = FALSE,
  cens = FALSE,
  right = NULL,
  left = NULL,
  mis = FALSE,
  prop = NULL,
  out = FALSE,
  eps = NULL,
  G = NULL,
  func = function(x) rowMeans(x, na.rm = T),
  sim = NULL
)
```

Arguments

tol	positive convergence tolerance ϵ . The ib procedure converges when $\ \hat{\theta}^{k+1} - \hat{\theta}^k\ _2/p < \epsilon$, where p is the dimension of θ .
maxit	integer representing the maximal number of iterations.
verbose	if TRUE, it prints some output in the console at each iteration.
seed	integer to set the seed (see Random).
H	integer representing the number of bootstrap estimates (see ib).
constraint	if TRUE (default), constraint for extra_param is used in the iterative procedure (see 'Details' of ib).
early_stop	if TRUE (default is FALSE), the iterative procedure stops as soon as there is no improvment in the minimization of the objective function (see 'Details' of ib).
cens	if TRUE the simulated responses are censored according to left and right values.
right	double for right-censoring (only used if cens=TRUE).
left	double for left-censoring (only used if cens=TRUE).
mis	if TRUE the simulated responses have missing data at random.
prop	double between 0 and 1 representing the proportion of missing data (only used if mis=TRUE).
out	if TRUE the simulated responses are also generated with a contamination mechanism.
eps	double between 0 and 1 representing the proportion of outliers in the data (only used if out=TRUE).
G	a function to generate outliers. It takes only a sample size as argument.
func	a function to reduce the H bootstrap estimates (rowwise). By default, the average is computed. The user can supply a function. One could imagine using other function such as the median or a trimmed mean.
sim	a user-defined function for simulating responses (see 'Details')

Details

sim allows the user to provide its own function for generating responses. Currently it is only supported for generalized linear models with the prototype 'fun(object, control, extra_param, ...)' (see [ib](#)).

Value

a list with components named as the arguments.

See Also

[ib](#), the iterative procedure for bias correction.

plot,Ib,ANY-method	<i>Method for plotting an object in class union "Ib"</i>
--------------------	--

Description

Method for plotting an object in class union "Ib"

Usage

```
## S4 method for signature 'Ib,ANY'  
plot(x, y = NULL, ...)
```

Arguments

x	an object of class union "Ib"
y	not used
...	further arguments to pass to plot

See Also

[Ib](#), [plot.lm](#)

predict,Ib-method	<i>Method for making predictions from an object in class union "Ib"</i>
-------------------	---

Description

Method for making predictions from an object in class union "Ib"

Usage

```
## S4 method for signature 'Ib'  
predict(object, ...)
```

Arguments

object	an object of class union "Ib"
...	further arguments to pass to predict

See Also

[Ib](#), [predict](#)

residuals,Ib-method	<i>Method for extracting residuals from an object in class union "Ib"</i>
---------------------	---

Description

Method for extracting residuals from an object in class union "Ib"

Usage

```
## S4 method for signature 'Ib'  
residuals(object, ...)
```

Arguments

object	an object of class union "Ib"
...	further arguments to pass to residuals

See Also

[Ib, residuals](#)

show,Ib-method	<i>Method for printing object in class union "Ib"</i>
----------------	---

Description

Method for printing object in class union "Ib"

Usage

```
## S4 method for signature 'Ib'  
show(object)
```

Arguments

object	an object of class union "Ib"
--------	-------------------------------

See Also

[Ib](#)

show,SummaryIb-method *Summarizing a fitted model corrected by the ib procedure*

Description

Method for printing a summary of class union [SummaryIb](#).

Usage

```
## S4 method for signature 'SummaryIb'
show(object)
```

Arguments

object a summary object of member of [SummaryIb](#)

See Also

[SummaryIb](#)

simulation *Generic for simulating from the object*

Description

Method for simulating responses from an object.

Usage

```
simulation(object, control = list(...), ...)

## S4 method for signature 'Ib'
simulation(object, control = list(...), ...)
```

Arguments

object an object of class union "Ib"

control a control list

... further argument to pass

Value

simulated responses.

Examples

```
## bootstrap poisson regression
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
pois_fit <- glm(counts ~ outcome + treatment, family = poisson())

## make 100 paramtric bootstrap replicates
boot_dist <- simulate(pois_fit, nsim = 100)
```

simulation,betareg-method

Simulation for a beta regression

Description

simulation method for class IbBetareg, see [Ib](#)

Usage

```
## S4 method for signature 'betareg'
simulation(object, control = list(...), extra = NULL, ...)
```

Arguments

object	an object of class IbBetareg, see Ib
control	a list of parameters for controlling the iterative procedure (see ibControl).
extra	NULL by default; extra parameters to pass to simulation.
...	further arguments

simulation,glm-method *Simulation for a Generalized Linear Model regression*

Description

simulation method for class IbGlm, see [Ib](#)

Usage

```
## S4 method for signature 'glm'
simulation(object, control = list(...), extra = NULL, ...)
```

Arguments

object	an object of class IbGlm, see Ib
control	a list of parameters for controlling the iterative procedure (see ibControl).
extra	NULL by default; extra parameters to pass to simulation.
...	further arguments

simulation,glmerMod-method

Simulation for linear mixed model regression

Description

simulation method for class IbGlmr, see [Ib](#)

Usage

```
## S4 method for signature 'glmerMod'
simulation(object, control = list(...), ...)
```

Arguments

object	an object of class IbGlmr, see Ib
control	a list of parameters for controlling the iterative procedure (see ibControl).
...	further arguments.

simulation,lm-method

Simulation for linear regression

Description

simulation method for class IbLm, see [Ib](#)

Usage

```
## S4 method for signature 'lm'
simulation(object, control = list(...), std = NULL, ...)
```

Arguments

object	an object of class IbLm, see Ib
control	a list of parameters for controlling the iterative procedure (see ibControl).
std	NULL by default; standard deviation to pass to simulation.
...	further arguments

simulation,lmerMod-method

Simulation for linear mixed model regression

Description

simulation method for class Ib, see [Ib](#)

Usage

```
## S4 method for signature 'lmerMod'
simulation(object, control = list(...), ...)
```

Arguments

object	an object of class Ib, see Ib
control	a list of parameters for controlling the iterative procedure (see ibControl).
...	further arguments.

simulation,negbin-method

Simulation for a negative binomial regression

Description

simulation method for class IbNegbin, see [Ib](#)

Usage

```
## S4 method for signature 'negbin'
simulation(object, control = list(...), extra = NULL, ...)
```

Arguments

object	an object of class IbNegbin, see Ib
control	a list of parameters for controlling the iterative procedure (see ibControl).
extra	NULL by default; extra parameters to pass to simulation.
...	further arguments

simulation,nls-method *Simulation for nonlinear regression*

Description

simulation method for class IbNls, see [Ib](#)

Usage

```
## S4 method for signature 'nls'  
simulation(object, control = list(...), std = NULL, ...)
```

Arguments

object	an object of class IbNls, see Ib
control	a list of parameters for controlling the iterative procedure (see ibControl).
std	NULL by default; standard deviation to pass to simulation.
...	further arguments

simulation,vglm-method *Simulation for vector generalized linear model regression*

Description

simulation method for class IbVglm, see [Ib](#)

Usage

```
## S4 method for signature 'vglm'  
simulation(object, control = list(...), extra_param = NULL, ...)
```

Arguments

object	an object of class IbVglm, see Ib
control	a list of parameters for controlling the iterative procedure (see ibControl).
extra_param	NULL by default; extra parameters to pass to simulation.
...	further arguments

summary,IbBetareg-method

Summarizing a beta regression fit corrected by the iterative bootstrap

Description

summary method for class [IbBetareg](#)

Usage

```
## S4 method for signature 'IbBetareg'  
summary(object, ...)
```

Arguments

object	an object of class IbBetareg
...	further arguments passed to <code>summary.betareg</code>

See Also

[summary.betareg](#)

summary,IbGlm-method

Summarizing a Generalized Linear Model regression fit corrected by the iterative bootstrap

Description

summary method for class [IbGlm](#)

Usage

```
## S4 method for signature 'IbGlm'  
summary(object, ...)
```

Arguments

object	an object of class IbGlm
...	further arguments passed to <code>summary.glm</code>

See Also

[summary.glm](#)

summary,IbGlmr-method

Summarizing a generalized linear mixed model regression fit corrected by the iterative bootstrap

Description

summary method for class [IbGlmr](#)

Usage

```
## S4 method for signature 'IbGlmr'
summary(object, ...)
```

Arguments

object	an object of class IbGlmr
...	further arguments passed to summary.merMod of lme4

summary,IbLm-method

Summarizing a linear regression fit corrected by the iterative bootstrap

Description

summary method for class [IbLm](#)

Usage

```
## S4 method for signature 'IbLm'
summary(object, ...)
```

Arguments

object	an object of class IbLm
...	further arguments passed to summary.lm

See Also

[summary.lm](#)

summary, IbLmer-method	<i>Summarizing a linear mixed model regression fit corrected by the iterative bootstrap</i>
------------------------	---

Description

summary method for class [IbLmer](#)

Usage

```
## S4 method for signature 'IbLmer'  
summary(object, ...)
```

Arguments

object	an object of class IbLmer
...	further arguments passed to <code>summary.merMod</code> of lme4

summary, IbNegbin-method	<i>Summarizing a negative binomial regression fits corrected by the iterative bootstrap</i>
--------------------------	---

Description

summary method for class [IbNegbin](#)

Usage

```
## S4 method for signature 'IbNegbin'  
summary(object, ...)
```

Arguments

object	an object of class IbNegbin
...	further arguments passed to <code>summary.negbin</code>

See Also

[summary.negbin](#)

summary,IbNls-method	<i>Summarizing a nonlinear regression fit corrected by the iterative bootstrap</i>
----------------------	--

Description

summary method for class [IbNls](#)

Usage

```
## S4 method for signature 'IbNls'  
summary(object, ...)
```

Arguments

object	an object of class IbNls
...	further arguments passed to summary.nls of stats

summary,IbVglm-method	<i>Summarizing a vector generalized linear model regression fit corrected by the iterative bootstrap</i>
-----------------------	--

Description

summary method for class [IbVglm](#)

Usage

```
## S4 method for signature 'IbVglm'  
summary(object, ...)
```

Arguments

object	an object of class IbVglm
...	further arguments passed to summary.merMod of VGAM

SummaryIb-class	Summaries for ib wrappers
-----------------	---------------------------

Description

S4 classes that wrap summary objects corresponding to the Ib* classes.
Members: [SummaryIbBetareg](#), [SummaryIbGlm](#), [SummaryIbLm](#), [SummaryIbLmer](#), [SummaryIbGlmer](#), [SummaryIbNegbin](#), [SummaryIbNls](#), [SummaryIbVglm](#).

Slots

summ The summary.* object.
ib_extra A list of extra metadata from ib.

See Also

[SummaryIb](#), [Ib](#)

vcov,Ib-method	Method for calculating covariance matrix from an object in class union "Ib"
----------------	---

Description

Method for calculating covariance matrix from an object in class union "Ib"

Usage

```
## S4 method for signature 'Ib'
vcov(object, ...)
```

Arguments

object	an object of class union "Ib"
...	further arguments to pass to vcov

See Also

[Ib](#), [vcov](#)

Index

- betareg, [9](#), [10](#)
- bootstrap, [3](#)
- coef, [4](#), [5](#)
- coef, Ib-method, [3](#)
- coefficients, [9](#)
- effects, [4](#)
- effects, Ib-method, [4](#)
- fitted, Ib-method, [4](#)
- fitted.values, [5](#)
- Gamma, [9](#)
- getEst, [5](#)
- getEst, Ib-method (getEst), [5](#)
- getExtra, [5](#), [5](#), [7](#), [13](#)
- getExtra, Ib-method (getExtra), [5](#)
- getIteration, [6](#)
- getIteration, Ib-method (getIteration), [6](#)
- getObject, [7](#), [13](#)
- getObject, Ib-method (getObject), [7](#)
- glm, [9](#), [10](#)
- glm.nb, [9](#), [10](#), [12](#)
- glmer, [9](#), [10](#)
- Ib, [4–7](#), [9](#), [13](#), [15](#), [16](#), [18–21](#), [26](#)
- Ib (Ib-class), [12](#)
- ib, [3](#), [7](#), [12–14](#)
- ib, betareg-method (ib), [7](#)
- ib, glm-method (ib), [7](#)
- ib, glmerMod-method (ib), [7](#)
- ib, lm-method (ib), [7](#)
- ib, lmerMod-method (ib), [7](#)
- ib, negbin-method, [12](#)
- ib, nls-method (ib), [7](#)
- ib, vglm-method (ib), [7](#)
- Ib-class, [12](#)
- IbBetareg, [12](#), [22](#)
- IbBetareg-class (Ib-class), [12](#)
- ibControl, [3](#), [8](#), [9](#), [12](#), [13](#), [18–21](#)
- IbGlm, [12](#), [22](#)
- IbGlm-class (Ib-class), [12](#)
- IbGlmer, [12](#), [23](#)
- IbGlmer-class (Ib-class), [12](#)
- IbLm, [12](#), [23](#)
- IbLm-class (Ib-class), [12](#)
- IbLmer, [12](#), [24](#)
- IbLmer-class (Ib-class), [12](#)
- IbNegbin, [12](#), [24](#)
- IbNegbin-class (Ib-class), [12](#)
- IbNls, [12](#), [25](#)
- IbNls-class (Ib-class), [12](#)
- IbVglm, [12](#), [25](#)
- IbVglm-class (Ib-class), [12](#)
- inverse.gaussian, [9](#)
- lm, [9](#), [10](#)
- lmer, [9](#), [10](#)
- nls, [9](#), [10](#)
- plot, Ib, ANY-method, [15](#)
- plot.lm, [15](#)
- predict, [15](#)
- predict, Ib-method, [15](#)
- quasi, [9](#)
- Random, [14](#)
- residuals, [16](#)
- residuals, Ib-method, [16](#)
- show, Ib-method, [16](#)
- show, SummaryIb-method, [17](#)
- simulate, [9](#)
- simulate.vlm, [9](#)
- simulation, [17](#)
- simulation, betareg-method, [18](#)
- simulation, glm-method, [18](#)
- simulation, glmerMod-method, [19](#)
- simulation, Ib-method (simulation), [17](#)

simulation, lm-method, [19](#)
simulation, lmerMod-method, [20](#)
simulation, negbin-method, [20](#)
simulation, nls-method, [21](#)
simulation, vglm-method, [21](#)
summary, IbBetareg-method, [22](#)
summary, IbGlm-method, [22](#)
summary, IbGlmer-method, [23](#)
summary, IbLm-method, [23](#)
summary, IbLmer-method, [24](#)
summary, IbNegbin-method, [24](#)
summary, IbNls-method, [25](#)
summary, IbVglm-method, [25](#)
summary.betareg, [22](#)
summary.glm, [22](#)
summary.lm, [23](#)
summary.negbin, [24](#)
SummaryIb, [13](#), [17](#), [26](#)
SummaryIb (SummaryIb-class), [26](#)
SummaryIb-class, [26](#)
SummaryIbBetareg, [26](#)
SummaryIbBetareg-class
 (SummaryIb-class), [26](#)
SummaryIbGlm, [26](#)
SummaryIbGlm-class (SummaryIb-class), [26](#)
SummaryIbGlmer, [26](#)
SummaryIbGlmer-class (SummaryIb-class),
 [26](#)
SummaryIbLm, [26](#)
SummaryIbLm-class (SummaryIb-class), [26](#)
SummaryIbLmer, [26](#)
SummaryIbLmer-class (SummaryIb-class),
 [26](#)
SummaryIbNegbin, [26](#)
SummaryIbNegbin-class
 (SummaryIb-class), [26](#)
SummaryIbNls, [26](#)
SummaryIbNls-class (SummaryIb-class), [26](#)
SummaryIbVglm, [26](#)
SummaryIbVglm-class (SummaryIb-class),
 [26](#)

vcov, [26](#)
vcov, Ib-method, [26](#)
vglm, [9](#), [10](#)