

# Package: **ib** (via r-universe)

November 3, 2024

**Type** Package

**Title** Bias Correction via Iterative Bootstrap

**Version** 0.2.0

**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](mailto: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

**LazyData** true

**RdMacros** Rdpack

**RoxygenNote** 7.1.2

**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

**Repository** <https://smac-group.r-universe.dev>

**RemoteUrl** <https://github.com/smac-group/ib>

**RemoteRef** HEAD

**RemoteSha** 0e1fb6994b45f28e1c9d2103bf909d5a68508c1e

## Contents

bootstrap	2
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	11
IbBetareg-class	12
ibControl	13
plot,Ib,ANY-method	14
predict,Ib-method	15
residuals,Ib-method	15
show,Ib-method	16
show,SummaryIb-method	16
simulation	17
simulation,betareg-method	17
simulation,glm-method	18
simulation,lm-method	18
simulation,lmerMod-method	19
simulation,negbin-method	19
simulation,nls-method	20
simulation,vglm-method	20
summary,IbBetareg-method	21
summary,IbGlm-method	21
summary,IbLm-method	22
summary,IbLmer-method	22
summary,IbNegbin-method	23
summary,IbNls-method	23
summary,IbVglm-method	24
SummaryIbBetareg-class	24
vcov,Ib-method	25
<b>Index</b>	<b>26</b>

---

bootstrap

*Parametric bootstrap*

---

### 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 `ibControl`.

**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

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

---

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

---

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

---

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

---

**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	<i>Accessor to an extra part in class union "Ib"</i>
----------	--

---

**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 '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 <code>ibControl</code> ).
<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 `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`  
`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)

## 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
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
## Not run:
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 <a href="#">ibControl</a> ).
extra_param	if TRUE, the bias of estimation of extra parameters is performed (see 'Details').
...	additional optional arguments (see 'Details').

---

IbBetareg-class	<i>An S4 class union for ib</i>
-----------------	---------------------------------

---

### Description

Members of the union are [IbBetareg](#), [IbGlm](#), [IbLm](#), [IbLmer](#), [IbNegbin](#), [IbNls](#), [IbVglm](#)

### Details

The ‘Functions’ section describes members of the class union.

### Value

Each member of the union has a slot with the initial object corrected by the `ib` (see [getObject](#)) and a second slot with extra meta data from `ib` (see [getExtra](#)).

### Functions

- `IbBetareg-class`: fitted model by `betareg` from **betareg**
- `IbGlm-class`: fitted model by `glm` from **stats**
- `IbLm-class`: fitted model by `lm` from **stats**
- `IbLmer-class`: fitted model by `lmer` from **lme4**
- `IbNegbin-class`: fitted model by `glm.nb` from **MASS**
- `IbNls-class`: fitted model by `nls` from **stats**
- `IbVglm-class`: fitted model by `vglm` from **VGAM**

### Author(s)

Samuel Orso

### See Also

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

---

 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

<code>tol</code>	positive convergence tolerance $\epsilon$ . The <code>ib</code> procedure converges when $\ \hat{\theta}^{k+1} - \hat{\theta}^k\ _2/p < \epsilon$ , where $p$ is the dimension of $\theta$ .
<code>maxit</code>	integer representing the maximal number of iterations.
<code>verbose</code>	if TRUE, it prints some output in the console at each iteration.
<code>seed</code>	integer to set the seed (see <a href="#">Random</a> ).
<code>H</code>	integer representing the number of bootstrap estimates (see <code>ib</code> ).
<code>constraint</code>	if TRUE (default), constraint for <code>extra_param</code> is used in the iterative procedure (see 'Details' of <code>ib</code> ).
<code>early_stop</code>	if TRUE (default is FALSE), the iterative procedure stops as soon as there is no improvement in the minimization of the objective function (see 'Details' of <code>ib</code> ).
<code>cens</code>	if TRUE the simulated responses are censored according to <code>left</code> and <code>right</code> values.
<code>right</code>	double for right-censoring (only used if <code>cens=TRUE</code> ).

<code>left</code>	double for left-censoring (only used if <code>cens=TRUE</code> ).
<code>mis</code>	if <code>TRUE</code> the simulated responses have missing data at random.
<code>prop</code>	double between 0 and 1 representing the proportion of missing data (only used if <code>mis=TRUE</code> ).
<code>out</code>	if <code>TRUE</code> the simulated responses are also generated with a contamination mechanism.
<code>eps</code>	double between 0 and 1 representing the proportion of outliers in the data (only used if <code>out=TRUE</code> ).
<code>G</code>	a function to generate outliers. It takes only a sample size as argument.
<code>func</code>	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.
<code>sim</code>	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`      *Method for plotting an object in class union "Ib"*

---

**Description**

Method for plotting an object in class union "Ib"

**Usage**

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

**Arguments**

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

**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      *Method for printing object in class union "Ib"*

---

**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	<i>Generic for simulating from the object</i>
------------	---

---

**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	<i>Simulation for a beta regression</i>
----------------------------	---

---

**Description**

simulation method for class [IbBetareg](#)

**Usage**

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

**Arguments**

object	an object of class <a href="#">IbBetareg</a>
control	a list of parameters for controlling the iterative procedure (see <a href="#">ibControl</a> ).
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](#)

**Usage**

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

**Arguments**

object	an object of class <a href="#">IbGlm</a>
control	a list of parameters for controlling the iterative procedure (see <a href="#">ibControl</a> ).
extra	NULL by default; extra parameters to pass to simulation.
...	further arguments

---

simulation,lm-method *Simulation for linear regression*

---

**Description**

simulation method for class [IbLm](#)

**Usage**

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

**Arguments**

object	an object of class <a href="#">IbLm</a>
control	a list of parameters for controlling the iterative procedure (see <a href="#">ibControl</a> ).
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 [IbLmer](#)

### Usage

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

### Arguments

object	an object of class <a href="#">IbLmer</a>
control	a list of parameters for controlling the iterative procedure (see <a href="#">ibControl</a> ).
...	further arguments.

---

simulation,negbin-method

*Simulation for a negative binomial regression*

---

### Description

simulation method for class [IbNegbin](#)

### Usage

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

### Arguments

object	an object of class <a href="#">IbNegbin</a>
control	a list of parameters for controlling the iterative procedure (see <a href="#">ibControl</a> ).
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](#)

### Usage

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

### Arguments

object	an object of class <a href="#">IbNls</a>
control	a list of parameters for controlling the iterative procedure (see <a href="#">ibControl</a> ).
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](#)

### Usage

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

### Arguments

object	an object of class <a href="#">IbVglm</a>
control	a list of parameters for controlling the iterative procedure (see <a href="#">ibControl</a> ).
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 `summary.betareg`

### 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 `summary.glm`

### See Also

[summary.glm](#)

---

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

---

### 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 `summary.merMod` of **lme4**

---

summary,IbNegbin-method

*Summarizing a negative binomial regression fits corrected by the iterative bootstrap*

---

### 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 `summary.negbin`

### See Also

[summary.negbin](#)

---

summary,IbNls-method

*Summarizing a nonlinear regression fit corrected by the iterative bootstrap*

---

### 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 *Summarizing a vector generalized linear model regression fit corrected by the iterative bootstrap*

---

### 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**

---

SummaryIbBetareg-class

*An S4 class union for summary*

---

### Description

Members of the union are [SummaryIbBetareg](#), [SummaryIbGlm](#), [SummaryIbLm](#), [SummaryIbLmer](#), [SummaryIbNegbin](#), [SummaryIbNls](#), [SummaryIbVglm](#) iterative bootstrap procedure

### Details

The ‘Functions’ section describes members of the class union.

### Functions

- [SummaryIbBetareg-class](#): summary of class `summary.betareg` from **betareg**
- [SummaryIbGlm-class](#): summary of class `summary.glm` from **stats**
- [SummaryIbLm-class](#): summary of class `summary.lm` from **stats**
- [SummaryIbLmer-class](#): summary of class `summary.merMod` from **lme4**
- [SummaryIbNegbin-class](#): summary of class `summary.negbin` from **MASS**
- [SummaryIbNls-class](#): summary of class `summary.nls` from **stats**
- [SummaryIbVglm-class](#): summary of class `summary.vglm` from **VGAM**

### Author(s)

Samuel Orso



---

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

---

**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, 8, 9  
bootstrap, 2  
coef, 3, 5  
coef, Ib-method, 3  
coefficients, 8  
effects, 4  
effects, Ib-method, 4  
fitted, Ib-method, 4  
fitted.values, 4  
Gamma, 8  
getEst, 5  
getEst, Ib-method (getEst), 5  
getExtra, 5, 5, 6, 12  
getExtra, Ib-method (getExtra), 5  
getIteration, 6  
getIteration, Ib-method (getIteration), 6  
getObject, 7, 12  
getObject, Ib-method (getObject), 7  
glm, 8, 9  
glm.nb, 8, 9, 11  
Ib, 3–7, 9, 15, 16, 25  
ib, 3, 7, 11, 13, 14  
ib, betareg-method (ib), 7  
ib, glm-method (ib), 7  
ib, lm-method (ib), 7  
ib, lmerMod-method (ib), 7  
ib, negbin-method, 11  
ib, nls-method (ib), 7  
ib, vglm-method (ib), 7  
Ib-class (IbBetareg-class), 12  
IbBetareg, 12, 17, 18, 21  
IbBetareg-class, 12  
ibControl, 3, 8, 11, 13, 18–20  
IbGlm, 12, 18, 21  
IbGlm-class (IbBetareg-class), 12  
IbLm, 12, 18, 22  
IbLm-class (IbBetareg-class), 12  
IbLmer, 12, 19, 22  
IbLmer-class (IbBetareg-class), 12  
IbNegbin, 12, 19, 23  
IbNegbin-class (IbBetareg-class), 12  
IbNls, 12, 20, 23  
IbNls-class (IbBetareg-class), 12  
IbVglm, 12, 20, 24  
IbVglm-class (IbBetareg-class), 12  
inverse.gaussian, 8  
lm, 8, 9  
lmer, 8, 9  
nls, 9  
plot, Ib, ANY-method, 14  
plot.lm, 15  
predict, 15  
predict, Ib-method, 15  
quasi, 8  
Random, 13  
residuals, 15  
residuals, Ib-method, 15  
show, Ib-method, 16  
show, SummaryIb-method, 16  
simulate, 8  
simulate.vlm, 9  
simulation, 17  
simulation, betareg-method, 17  
simulation, glm-method, 18  
simulation, Ib-method (simulation), 17  
simulation, lm-method, 18  
simulation, lmerMod-method, 19  
simulation, negbin-method, 19  
simulation, nls-method, 20  
simulation, vglm-method, 20  
summary, IbBetareg-method, 21

summary, IbGlm-method, [21](#)  
summary, IbLm-method, [22](#)  
summary, IbLmer-method, [22](#)  
summary, IbNegbin-method, [23](#)  
summary, IbNls-method, [23](#)  
summary, IbVglm-method, [24](#)  
summary.betareg, [21](#)  
summary.glm, [21](#)  
summary.lm, [22](#)  
summary.negbin, [23](#)  
SummaryIb, [16](#)  
SummaryIb-class  
    (SummaryIbBetareg-class), [24](#)  
SummaryIbBetareg, [24](#)  
SummaryIbBetareg-class, [24](#)  
SummaryIbGlm, [24](#)  
SummaryIbGlm-class  
    (SummaryIbBetareg-class), [24](#)  
SummaryIbLm, [24](#)  
SummaryIbLm-class  
    (SummaryIbBetareg-class), [24](#)  
SummaryIbLmer, [24](#)  
SummaryIbLmer-class  
    (SummaryIbBetareg-class), [24](#)  
SummaryIbNegbin, [24](#)  
SummaryIbNegbin-class  
    (SummaryIbBetareg-class), [24](#)  
SummaryIbNls, [24](#)  
SummaryIbNls-class  
    (SummaryIbBetareg-class), [24](#)  
SummaryIbVglm, [24](#)  
SummaryIbVglm-class  
    (SummaryIbBetareg-class), [24](#)  
  
vcov, [25](#)  
vcov, Ib-method, [25](#)  
vglm, [9](#)