

# Package ‘abe’

April 3, 2025

**Type** Package

**Title** Augmented Backward Elimination

**Version** 5.1.2

**Date** 2025-4-1

**Author** Rok Blagus [aut, cre],  
Sladana Babic [ctb],  
Daniela Dunkler [ctb],  
Georg Heinze [ctb],  
Gregor Steiner [ctb]

**Maintainer** Rok Blagus <rok.blagus@mf.uni-lj.si>

**Description** Performs augmented backward elimination and checks the stability of the obtained model. Augmented backward elimination combines significance or information based criteria with the change in estimate to either select the optimal model for prediction purposes or to serve as a tool to obtain a practically sound, highly interpretable model. More details can be found in Dunkler et al. (2014) <[doi:10.1371/journal.pone.0113677](https://doi.org/10.1371/journal.pone.0113677)>.

**License** GPL-3

**Depends** R (>= 4.1.0)

**RoxygenNote** 7.3.2

**Encoding** UTF-8

**Imports** ggplot2 (>= 3.4.0), reshape2 (>= 1.4.0), tidytext (>= 0.4.0),  
survival (>= 3.4-0), foreach (>= 1.5.0), lifecycle (>= 1.0.0)

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2025-04-03 10:50:06 UTC

## Contents

abe . . . . .	2
---------------	---

abe.resampling . . . . .	5
pie.abe . . . . .	10
plot.abe . . . . .	11
print.abe . . . . .	13
summary.abe . . . . .	15

<b>Index</b>	<b>18</b>
--------------	-----------

---

abe	<i>Augmented Backward Elimination</i>
-----	---------------------------------------

---

## Description

Function 'abe' performs Augmented Backward Elimination where variable selection is based on the change-in-estimate and significance or information criteria as presented in [Dunkler et al. (2014)](doi:10.1371/journal.pone.0141111). It can also make a backward elimination based on significance or information criteria only by turning off the change-in-estimate criterion.

## Usage

```

abe(
  fit,
  data = NULL,
  include = NULL,
  active = NULL,
  tau = 0.05,
  exact = FALSE,
  criterion = c("alpha", "AIC", "BIC"),
  alpha = 0.2,
  type.test = c("Chisq", "F", "Rao", "LRT"),
  type.factor = NULL,
  verbose = TRUE,
  ...
)

```

## Arguments

fit	An object of class "lm", "glm", "logistf", "coxph", or "survreg" representing the fit. Note, the functions should be fitted with argument 'x=TRUE' and 'y=TRUE' (or 'model=TRUE' for "logistf" objects).
data	data frame used when fitting the object 'fit'.
include	a vector containing the names of variables that will be included in the final model. These variables are used as only passive variables during modeling. *These variables might be exposure variables of interest or known confounders.* They will never be dropped from the working model in the selection process, but they will be used passively in evaluating change-in-estimate criteria of other variables. Note, variables which are not specified as include or active in the model fit are assumed to be active and passive variables.

active	a vector containing the names of active variables. These <i>*less important explanatory variables*</i> will only be used as active, but not as passive variables when evaluating the change-in-estimate criterion.
tau	Value that specifies the threshold of the relative change-in-estimate criterion. Default is set to 0.05.
exact	Logical, specifies if the method will use exact change-in-estimate or its approximation. Default is set to FALSE, which means that the method will use the approximation proposed by Dunkler et al. (2014). Note, setting to TRUE can severely slow down the algorithm, but setting to FALSE can in some cases, i.e., if dummy variables of a factor are evaluated together, lead to a poor approximation of the change-in-estimate criterion. See details.
criterion	String that specifies the strategy to select variables for the black list. Currently supported options are significance level "alpha", Akaike information criterion "AIC" and Bayesian information criterion "BIC". If you are using significance level, you have to specify the value of 'alpha' (see parameter 'alpha') and the type of the test statistic (see parameter 'type.test'). Default is set to "alpha".
alpha	Value that specifies the level of significance as explained above. Default is set to 0.2.
type.test	String that specifies which test should be performed in case the 'criterion = "alpha"'. Possible values are "F" and "Chisq" (default) for class "lm", "Rao", "LRT", "Chisq" (default), "F" for class "glm" and "Chisq" for class "coxph". See also <a href="#">drop1</a> .
type.factor	String that specifies how to treat factors, see details, possible values are "factor" and "individual".
verbose	Logical that specifies if the variable selection process should be printed. This can severely slow down the algorithm. Default is set to TRUE.
...	Further arguments. Currently, this is primarily used to warn users about arguments that are no longer supported.

## Details

Using the default settings 'abe' will perform augmented backward elimination based on significance. The level of significance will be set to 0.2. All variables will be treated as "passive or active". Approximated change-in-estimate will be used. Threshold of the relative change-in-estimate criterion will be 0.05. Setting tau to a very large number (e.g. 'Inf') turns off the change-in-estimate criterion, and ABE will only perform backward elimination. Specifying "alpha" = 0 will include variables only because of the change-in-estimate criterion, as then variables are not safe from exclusion because of their p-values. Specifying "alpha" = 1 will always include all variables.

When using 'type.factor="individual"' each dummy variable of a factor is treated as an individual explanatory variable, hence only this dummy variable can be removed from the model. Use sensible coding for the reference group. Using 'type.factor="factor"' will look at the significance of removing all dummy variables of the factor and can drop the entire variable from the model. If 'type.factor="factor"' then 'exact' should be set to 'TRUE' to avoid poor approximations.

In earlier versions, abe used to include an exp.beta argument. This is not supported anymore. Instead, the function now uses the exponential change-in-estimate for logistic, Cox, and parametric survival models only.

**Value**

An object of class `"lm"`, `"glm"`, `"coxph"`, or `"survreg"` representing the model chosen by `abe` method.

**Author(s)**

Rok Blagus, <rok.blagus@mf.uni-lj.si>

Daniela Dunkler

Gregor Steiner

Sladana Babic

**References**

Daniela Dunkler, Max Plischke, Karen Lefondre, and Georg Heinze. Augmented Backward Elimination: A Pragmatic and Purposeful Way to Develop Statistical Models. *PloS One*, 9(11):e113677, 2014, [doi:](doi:10.1371/journal.pone.0113677).

**See Also**

[abe.resampling](#), [lm](#), [glm](#) and [coxph](#)

**Examples**

```
# simulate some data:

set.seed(1)
n = 100
x1 <- runif(n)
x2 <- runif(n)
x3 <- runif(n)
y <- -5 + 5 * x1 + 5 * x2 + rnorm(n, sd = 5)
dd <- data.frame(y, x1, x2, x3)

# fit a simple model containing all variables
fit1 <- lm(y ~ x1 + x2 + x3, x = TRUE, y = TRUE, data = dd)

# perform ABE with "x1" as only passive and "x2" as only active
# using the exact change in the estimate of 5% and significance
# using 0.2 as a threshold
abe.fit <- abe(fit1, data = dd, include = "x1", active = "x2",
tau = 0.05, exact = TRUE, criterion = "alpha", alpha = 0.2,
type.test = "Chisq", verbose = TRUE)

summary(abe.fit)

# similar example, but turn off the change-in-estimate and perform
# only backward elimination

be.fit <- abe(fit1, data = dd, include = "x1", active = "x2",
tau = Inf, exact = TRUE, criterion = "alpha", alpha = 0.2,
```

```

type.test = "Chisq", verbose = TRUE)

summary(be.fit)

# an example with the model containing categorical covariates:
dd$x4 <- rbinom(n, size = 3, prob = 1/3)
dd$y1 <- -5 + 5 * x1 + 5 * x2 + rnorm(n, sd = 5)
fit2 <- lm(y1 ~ x1 + x2 + factor(x4), x = TRUE, y = TRUE, data = dd)

# treat "x4" as a single covariate: perform ABE as in abe.fit

abe.fit.fact <- abe(fit2, data = dd, include = "x1", active = "x2",
tau = 0.05, exact = TRUE, criterion = "alpha", alpha = 0.2,
type.test = "Chisq", verbose = TRUE, type.factor = "factor")

summary(abe.fit.fact)

# treat each dummy of "x3" as a separate covariate: perform ABE as in abe.fit

abe.fit.ind <- abe(fit2, data = dd, include = "x1", active = "x2",
tau = 0.05, exact = TRUE, criterion = "alpha", alpha = 0.2,
type.test = "Chisq", verbose = TRUE, type.factor = "individual")

summary(abe.fit.ind)

```

---

abe.resampling

*Resampled Augmented Backward Elimination*


---

## Description

Performs Augmented backward elimination on re-sampled data sets using different bootstrap and re-sampling techniques.

## Usage

```

abe.resampling(
  fit,
  data = NULL,
  include = NULL,
  active = NULL,
  tau = 0.05,
  exact = FALSE,
  criterion = c("alpha", "AIC", "BIC"),
  alpha = 0.2,
  type.test = c("Chisq", "F", "Rao", "LRT"),
  type.factor = NULL,
  num.resamples = 100,
  type.resampling = c("Wallisch2021", "bootstrap", "mn.bootstrap", "subsampling"),
  prop.sampling = 0.5,

```

```

save.out = c("minimal", "complete"),
parallel = FALSE,
seed = NULL,
...
)

```

## Arguments

<code>fit</code>	An object of class <code>"lm"</code> , <code>"glm"</code> , <code>"logistf"</code> , <code>"coxph"</code> , or <code>"survreg"</code> representing the fit. Note, the functions should be fitted with argument <code>'x=TRUE'</code> and <code>'y=TRUE'</code> (or <code>'model=TRUE'</code> for <code>"logistf"</code> objects).
<code>data</code>	data frame used when fitting the object <code>'fit'</code> .
<code>include</code>	a vector containing the names of variables that will be included in the final model. These variables are used as passive variables during modeling. These variables might be exposure variables of interest or known confounders. They will never be dropped from the working model in the selection process, but they will be used passively in evaluating change-in-estimate criteria of other variables. Note, variables which are not specified as include or active in the model fit are assumed to be active and passive variables.
<code>active</code>	a vector containing the names of active variables. These less important explanatory variables will only be used as active, but not as passive variables when evaluating the change-in-estimate criterion.
<code>tau</code>	Value that specifies the threshold of the relative change-in-estimate criterion. Default is set to 0.05.
<code>exact</code>	Logical, specifies if the method will use exact change-in-estimate or approximated. Default is set to <code>FALSE</code> , which means that the method will use approximation proposed by Dunkler et al. Note, setting to <code>TRUE</code> can severely slow down the algorithm, but setting to <code>FALSE</code> can in some cases lead to a poor approximation of the change-in-estimate criterion.
<code>criterion</code>	String that specifies the strategy to select variables for the blacklist. Currently supported options are significance level <code>'alpha'</code> , Akaike information criterion <code>'AIC'</code> and Bayesian information criterion <code>'BIC'</code> . If you are using significance level, in that case you have to specify the value of <code>'alpha'</code> (see parameter <code>'alpha'</code> ). Default is set to <code>"alpha"</code> .
<code>alpha</code>	Value that specifies the level of significance as explained above. Default is set to 0.2.
<code>type.test</code>	String that specifies which test should be performed in case the <code>'criterion = "alpha"</code> . Possible values are <code>"F"</code> and <code>"Chisq"</code> (default) for class <code>"lm"</code> , <code>"Rao"</code> , <code>"LRT"</code> , <code>"Chisq"</code> (default), <code>"F"</code> for class <code>"glm"</code> and <code>"Chisq"</code> for class <code>"coxph"</code> . See also <a href="#">drop1</a> .
<code>type.factor</code>	String that specifies how to treat factors, see details, possible values are <code>"factor"</code> and <code>"individual"</code> .
<code>num.resamples</code>	number of resamples.
<code>type.resampling</code>	String that specifies the type of resampling. Possible values are <code>"Wallisch2021"</code> , <code>"bootstrap"</code> , <code>"mn.bootstrap"</code> , <code>"subsampling"</code> . Default is set to <code>"Wallisch2021"</code> . See details.

prop.sampling	Sampling proportion. Only applicable for <code>'type.boot="mn.bootstrap"'</code> and <code>'type.boot="subsampling"'</code> , defaults to 0.5. See details.
save.out	String that specifies if only the minimal output of the refitted models ( <code>'save.out="minimal"'</code> ) or the entire object ( <code>'save.out="complete"'</code> ) is to be saved. Defaults to <code>"minimal"</code>
parallel	Logical, specifies if the calculations should be run in parallel <code>'TRUE'</code> or not <code>'FALSE'</code> . Defaults to <code>'FALSE'</code> . See details.
seed	Numeric, a random seed to be used to form re-sampled datasets. Defaults to <code>'NULL'</code> . Can be used to assure complete reproducibility of the results, see Examples.
...	Further arguments. Currently, this is primarily used to warn users about arguments that are no longer supported.

### Details

`'type.resampling'` can be `'bootstrap'` (n observations drawn from the original data with replacement), `'mn.bootstrap'` (m out of n observations drawn from the original data with replacement), `'subsampling'` (m out of n observations drawn from the original data without replacement, where m is `'prop.sampling*n'`) and `'Wallisch2021'`. When using `'Wallisch2021'` the resampling is done twice: first time using bootstrap (these results are contained in `'models'`) and the second time using resampling with `'prop.sampling'` equal to 0.5 (these results are contained in `'models.wallisch'`); see Wallisch et al. (2021).

When using `'parallel=TRUE'` parallel backend must be registered before using `'abe.resampling'`. The parallel backends available will be system-specific; see `[foreach()]` for more details.

In earlier versions, `abe` used to include an `exp.beta` argument. This is not supported anymore. Instead, the function now uses the exponential change in estimate for logistic and Cox models only.

### Value

an object of class `'abe'` for which `'summary'`, `'plot'` and `'pie.abe'` functions are available. A list with the following elements:

`'coefficients'` a matrix of coefficients of the final models obtained after performing ABE on re-sampled datasets; if using `'type.resampling="Wallisch2021"'`, these models are obtained by using bootstrap.

`'coefficients.wallisch'` if using `'type.resampling="Wallisch2021"'` the coefficients of the final models obtained after performing ABE using resampling with `'prop.sampling'` equal to 0.5; `'NULL'` when using any other option in `'type.resampling'`.

`'models'` the final models obtained after performing ABE on re-sampled datasets, each object in the list is of the same class as `'fit'`; if using `'type.resampling="Wallisch2021"'`, these models are obtained by using bootstrap. These are only returned if `'save.out = "complete"'`.

`'models.wallisch'` similar as `'models'`; if using `'type.resampling="Wallisch2021"'` the coefficients and terms of the final models obtained after performing ABE using resampling with `'prop.sampling'` equal to 0.5; `'NULL'` when using any other option in `'type.resampling'`. These are only returned if `'save.out = "complete"'`.

`'model.parameters'` a dataframe of alpha and tau values corresponding to the resampled models.

‘num.boot’ number of resampled datasets  
 ‘criterion’ criterion used when constructing the black-list  
 ‘all.vars’ a list of variables used when estimating ‘fit’  
 ‘fit.global’ the initial model. In earlier versions of the package this parameter was called ‘fit.or’.  
 ‘misc’ the parameters of the call to ‘abe.resampling’  
 ‘id’ the rows of the data which were used when refitting the model; the list with elements ‘id1’ (the rows used to refit the model; when ‘type.resampling="Wallisch2021"' these are based on bootstrap) and ‘id2’ (‘NULL’ unless when ‘type.resampling="Wallisch2021"' in which case these are the rows used to refit the models based on subsampling)

### Author(s)

Rok Blagus, <rok.blagus@mf.uni-lj.si>  
 Daniela Dunkler  
 Sladana Babic

### References

Daniela Dunkler, Max Plischke, Karen Lefondre, and Georg Heinze. Augmented Backward Elimination: A Pragmatic and Purposeful Way to Develop Statistical Models. *PloS One*, 9(11):e113677, 2014, [doi:](doi:10.1371/journal.pone.0113677).

Riccardo De Bin, Silke Janitza, Willi Sauerbrei and Anne-Laure Boulesteix. Subsampling versus Bootstrapping in Resampling-Based Model Selection for Multivariable Regression. *Biometrics* 72, 272-280, 2016, [doi:](doi:10.1111/biom.12381).

Wallisch Christine, Dunkler Daniela, Rauch Geraldine, de Bin Ricardo, Heinze Georg. Selection of Variables for Multivariable Models: Opportunities and Limitations in Quantifying Model Stability by Resampling. *Statistics in Medicine* 40:369-381, 2021, [doi:](doi:10.1002/sim.8779).

### See Also

[abe](#), [summary.abe](#), [print.abe](#), [plot.abe](#), [pie.abe](#)

### Examples

```
# simulate some data and fit a model

set.seed(1)
n = 100
x1 <- runif(n)
x2 <- runif(n)
x3 <- runif(n)
y<- -5 + 5 * x1 + 5 * x2 + rnorm(n, sd = 5)
dd <- data.frame(y = y, x1 = x1, x2 = x2, x3 = x3)
fit <- lm(y ~ x1 + x2 + x3, x = TRUE, y = TRUE, data = dd)

# use ABE on 10 re-samples considering different
# change-in-estimate thresholds and significance levels
```



```

fit.resample1 <- abe.resampling(fit, data = dd, include = "x1",
active = "x2", tau = c(0.05, 0.1), exact = TRUE,
criterion = "alpha", alpha = c(0.2, 0.05), type.test = "Chisq",
num.resamples = 10, type.resampling = "Wallisch2021")

names(summary(fit.resample1))
summary(fit.resample1)$var.rel.frequencies
summary(fit.resample1)$model.rel.frequencies
summary(fit.resample1)$var.coefs[1]
summary(fit.resample1)$pair.rel.frequencies[1]
print(fit.resample1)

# use ABE on 10 bootstrap re-samples considering different
# change-in-estimate thresholds and significance levels

fit.resample2 <- abe.resampling(fit, data = dd, include = "x1",
active = "x2", tau = c(0.05, 0.1), exact = TRUE,
criterion = "alpha", alpha = c(0.2, 0.05), type.test = "Chisq",
num.resamples = 10, type.resampling = "bootstrap")

summary(fit.resample2)

# use ABE on 10 subsamples randomly selecting 50% of subjects
# considering different change-in-estimate thresholds and
# significance levels

fit.resample3 <- abe.resampling(fit, data = dd, include = "x1",
active = "x2", tau = c(0.05,0.1), exact = TRUE,
criterion = "alpha", alpha = c(0.2, 0.05), type.test = "Chisq",
num.resamples = 10, type.resampling = "subsampling", prop.sampling = 0.5)

summary(fit.resample3)

#Assure reproducibility of the results

fit.resample.1 <- abe.resampling(fit, data = dd, include = "x1",
active = "x2", tau = c(0.05, 0.1), exact = TRUE,
criterion = "alpha", alpha = c(0.2, 0.05), type.test = "Chisq",
num.resamples = 10, type.resampling = "Wallisch2021")

fit.resample.2 <- abe.resampling(fit, data = dd, include = "x1",
active = "x2", tau = c(0.05, 0.1), exact = TRUE,
criterion = "alpha", alpha = c(0.2, 0.05), type.test = "Chisq",
num.resamples = 10, type.resampling = "Wallisch2021")

#since different seeds are used, fit.resample.1 and fit.resample.2 give different results

fit.resample.3 <- abe.resampling(fit, data = dd, include = "x1",
active = "x2", tau = c(0.05, 0.1), exact = TRUE,
criterion = "alpha", alpha = c(0.2, 0.05), type.test = "Chisq",
num.resamples = 10, type.resampling = "Wallisch2021", seed = 87982)

fit.resample.4 <- abe.resampling(fit, data = dd, include = "x1",

```

```

active = "x2", tau = c(0.05, 0.1), exact = TRUE,
criterion = "alpha", alpha = c(0.2, 0.05), type.test = "Chisq",
num.resamples = 10, type.resampling = "Wallisch2021", seed = 87982)

#now fit.resample.3 and fit.resample.4 give exactly the same results

#' Example to run parallel computation on windows, using all but 2 cores

#library(doParallel)
#N_CORES <- detectCores()
#c1 <- makeCluster(N_CORES-2)
#registerDoParallel(c1)
#fit.resample <- abe.resampling(fit, data = dd, include = "x1", active = "x2",
#tau = c(0.05, 0.1), exact = TRUE, criterion = "alpha", alpha = c(0.2, 0.05),
#type.test = "Chisq", num.resamples = 50, type.resampling = "Wallisch2021")
#stopCluster(c1)

```

---

pie.abe

*Pie Function*

---

## Description

Pie function for the resampled/bootstrapped version of ABE. Plots a pie chart of the model frequencies for specified values of ‘alpha’ and ‘tau’.

## Usage

```
pie.abe(x, alpha = NULL, tau = NULL, labels = NA, ...)
```

## Arguments

x	an object of class “abe”, an object returned by a call to [abe.resampling()]
alpha	values of alpha for which the plot is to be made (can be a vector of length >1)
tau	values of tau for which the plot is to be made (can be a vector of length >1)
labels	plot labels, defaults to NA, i.e. no labels are plotted
...	Arguments to be passed to methods, such as graphical parameters (see [pie()], [barplot()], [hist()]).

## Details

When using ‘type.resampling=“Wallisch2021”‘ the plot is based on subsampling with sampling proportion equal to 0.5, otherwise as specified in ‘type.resampling’.

## Author(s)

Rok Blagus, <rok.blagus@mf.uni-lj.si>

Sladana Babic

**See Also**

[abe.resampling](#), [summary.abe](#), [plot.abe](#)

**Examples**

```

set.seed(10)
n = 100
x1 <- runif(n)
x2 <- runif(n)
x3 <- runif(n)
y <- -5 + 5 * x1 + 5 * x2 + rnorm(n, sd = 5)
dd <- data.frame(y = y, x1 = x1, x2 = x2, x3 = x3)
fit <- lm(y ~ x1 + x2 + x3, x = TRUE, y = TRUE, data = dd)

fit.resample <- abe.resampling(fit, data = dd, include = "x1", active = "x2",
tau = c(0.05, 0.1), exact = TRUE, criterion = "alpha", alpha = c(0.2, 0.05),
type.test = "Chisq", num.resamples = 50, type.resampling = "Wallisch2021")

pie.abe(fit.resample, alpha = 0.2, tau = 0.1)

fit.resample <- abe.resampling(fit, data = dd, include = "x1", active = "x2",
tau = c(0.05, 0.1), exact=TRUE, criterion = "alpha", alpha = c(0.2, 0.05),
type.test = "Chisq", num.resamples = 50, type.resampling = "subsampling")

pie.abe(fit.resample, alpha = 0.2, tau = 0.1)

```

---

plot.abe

*Plot Function*

---

**Description**

Plot function for the resampled/bootstrapped version of ABE.

**Usage**

```

## S3 method for class 'abe'
plot(
  x,
  type.plot = c("coefficients", "variables", "models", "stability", "pairwise"),
  alpha = NULL,
  tau = NULL,
  variable = NULL,
  type.stability = c("alpha", "tau"),
  pval = 0.01,
  ...
)

```

**Arguments**

<code>x</code>	an object of class <code>"abe"</code> , an object returned by a call to <code>[abe.resampling()]</code>
<code>type.plot</code>	string which specifies the type of the plot. See details.
<code>alpha</code>	values of alpha for which the plot is to be made (can be a vector of length >1)
<code>tau</code>	values of tau for which the plot is to be made (can be a vector of length >1)
<code>variable</code>	variables for which the plot is to be made (can be a vector of length >1)
<code>type.stability</code>	string which specifies the type of stability plot. See details.
<code>pval</code>	significance level to be used to determine a significant deviation from the expected pairwise inclusion frequency under independence (default 0.01). Only relevant if <code>'type.plot="pairwise"'</code> .
<code>...</code>	Arguments to be passed to methods, such as graphical parameters.

**Details**

When using `'type.plot="coefficients"'` the function plots a histogram of the estimated regression coefficients for the specified variables, alpha(s) and tau(s) obtained from different re-sampled datasets. When the variable is not included in the final model, its regression coefficient is set to zero. When using `'type.resampling="Wallisch2021"'` the plot is based on bootstrap, otherwise as specified in `'type.resampling'`.

When using `type.plot="variables"` the function plots a barplot of the relative inclusion frequencies of the specified variables, for the specified values of alpha and tau. When using `'type.resampling="Wallisch2021"'` the plot is based on subsampling with sampling proportion equal to 0.5, otherwise as specified in `'type.resampling'`.

When using `type.plot="models"` the function plots a barplot of the relative frequencies of the final models for specified alpha(s) and tau(s). When using `'type.resampling="Wallisch2021"'` the plot is based on subsampling with sampling proportion equal to 0.5, otherwise as specified in `'type.resampling'`.

When using `'type.plot="stability"'` the function plots variable inclusion frequencies for each value of alpha. `'type.stability'` specifies if inclusion frequencies should be plotted as a function of alpha (default) or tau.

When using `'type.plot="pairwise"'` the function plots a heatmap of differences between observed pairwise inclusion frequencies and the expected pairwise inclusion frequencies under independence. A high value indicates overselection, i.e. the pair of variables is selected together more often than expected under independence. Selection frequencies (in

**Author(s)**

Rok Blagus, <rok.blagus@mf.uni-lj.si>

Sladana Babic

Daniela Dunkler

Gregor Steiner

**See Also**

[abe.resampling](#), [summary.abe](#), [pie.abe](#)

**Examples**

```

set.seed(1)
n=100
x1<-runif(n)
x2<-runif(n)
x3<-runif(n)
y<--5+5*x1+5*x2+ rnorm(n,sd=5)
dd<-data.frame(y=y,x1=x1,x2=x2,x3=x3)
fit<-lm(y~x1+x2+x3,x=TRUE,y=TRUE,data=dd)

fit.resample<-abe.resampling(fit,data=dd,include="x1",active="x2",
tau=c(0.05,0.1),exact=TRUE,
criterion="alpha",alpha=c(0.2,0.05),type.test="Chisq",
num.resamples=50,type.resampling="Wallisch2021")

plot(fit.resample,type.plot="coefficients",
alpha=0.2,tau=0.1,variable=c("x1","x3"),
col="light blue")

plot(fit.resample,type.plot="variables",
alpha=0.2,tau=0.1,variable=c("x1","x2","x3"),
col="light blue",horiz=TRUE,las=1)

par(mar=c(4,6,4,2))
plot(fit.resample,type.plot="models",
alpha=0.2,tau=0.1,col="light blue",horiz=TRUE,las=1)

fit.resample<-abe.resampling(fit,data=dd,include="x1",active="x2",
tau=c(0.05,0.1),exact=TRUE,
criterion="alpha",alpha=c(0.2,0.05),type.test="Chisq",
num.resamples=50,type.resampling="bootstrap")

plot(fit.resample,type.plot="coefficients",
alpha=0.2,tau=0.1,variable=c("x1","x3"),
col="light blue")

fit.resample<-abe.resampling(fit,data=dd,include="x1",active="x2",
tau=c(0.05,0.1),exact=TRUE,
criterion="alpha",alpha=c(0.2,0.05),type.test="Chisq",
num.resamples=50,type.resampling="subsampling")

plot(fit.resample,type.plot="variables",
alpha=0.2,tau=0.1,variable=c("x1","x2","x3"),
col="light blue",horiz=TRUE,las=1)

par(mar=c(4,6,4,2))
plot(fit.resample,type.plot="models",
alpha=0.2,tau=0.1,col="light blue",horiz=TRUE,las=1)

```

## Description

Prints a summary table of a bootstrapped/resampled version of ABE. The table displays the relative inclusion frequencies of the covariates from the initial model, the coefficient estimates and standard errors from the initial model (model with all covariates), the selected model, resampled median and percentiles for the estimates of the regression coefficients for each variable from the initial model, root mean squared difference ratio (RMSD) and relative bias conditional on selection (RBCS), see ‘details’.

## Usage

```
## S3 method for class 'abe'
print(
  x,
  type = c("coefficients", "coefficients reporting", "models"),
  models.n = NULL,
  conf.level = 0.95,
  alpha = NULL,
  tau = NULL,
  digits = 3,
  ...
)
```

## Arguments

<code>x</code>	an object of class “abe”, an object returned by a call to [abe.resampling()]
<code>type</code>	the type of the output. ‘type = “coefficients”‘ prints summary statistics for each coefficient, ‘type = “coefficients reporting”‘ prints a reduced version of the coefficient statistics, and ‘type = “models”‘ reports model selection frequencies.
<code>models.n</code>	controls the number of models printed if ‘type = “models”‘. See details.
<code>conf.level</code>	the confidence level, defaults to 0.95, see ‘details’
<code>alpha</code>	the alpha value for which the output is to be printed, defaults to ‘NULL’
<code>tau</code>	the tau value for which the output is to be printed, defaults to ‘NULL’
<code>digits</code>	integer, indicating the number of digits to display in the table. Defaults to 2
<code>...</code>	additional arguments affecting the summary produced.

## Details

When using ‘type.resampling=“Wallisch2021”‘ in a call to [abe.resampling()], the results for the relative inclusion frequencies of the covariates from the initial model are based on subsampling with sampling proportion equal to 0.5 and the other results are based on bootstrap as suggested by Wallisch et al. (2021); otherwise all the results are obtained by using the method as specified in ‘type.resampling’. Parameter ‘conf.level’ defines the lower and upper quantile of the bootstrapped/resampled distribution such that equal proportion of values are smaller and larger than the lower and the upper quantile, respectively.

If ‘type = “models”‘, the ‘models.n’ parameter controls the number of models printed. One option is to directly specify the number of models to return (i.e. an integer larger than 1). Alternatively, if

‘models.n’ is set to a number less than (or equal to) 1, the number of models returned is such that the cumulative frequency attains that value. By default (‘models.n = NULL’), the top 20 models or all models up to a cumulative frequency of 0.8, whichever is shorter, are returned. The selected model is marked with an asterisk. If it is not among the printed models, it is added as the last model.

**Author(s)**

Rok Blagus, <rok.blagus@mf.uni-lj.si>

Sladana Babic

Daniela Dunkler

Gregor Steiner

**References**

Wallisch C, Dunkler D, Rauch G, de Bin R, Heinze G. Selection of variables for multivariable models: Opportunities and limitations in quantifying model stability by resampling. *Statistics in Medicine* 40:369-381, 2021.

**See Also**

[abe.resampling](#), [summary.abe](#), [plot.abe](#), [pie.abe](#)

**Examples**

```
set.seed(100)
n = 100
x1 <- runif(n)
x2 <- runif(n)
x3 <- runif(n)
y<- -5 + 5 * x1 + 5 * x2 + rnorm(n, sd = 5)
dd <- data.frame(y = y, x1 = x1, x2 = x2, x3 = x3)
fit <- lm(y ~ x1 + x2 + x3, x = TRUE, y = TRUE, data= dd)

fit.resample <- abe.resampling(fit, data = dd, include = "x1", active = "x2",
tau = c(0.05, 0.1), exact = TRUE, criterion = "alpha", alpha = c(0.2, 0.05),
type.test = "Chisq", num.resamples = 50, type.resampling = "Wallisch2021")

print(fit.resample, conf.level = 0.95, alpha = 0.2, tau = 0.05)
```

---

summary.abe

*Summary Function*

---

**Description**

makes a summary of a resampled version of ABE

**Usage**

```
## S3 method for class 'abe'
summary(
  object,
  conf.level = 0.95,
  pval = 0.01,
  alpha = NULL,
  tau = NULL,
  models.n = NULL,
  ...
)
```

**Arguments**

object	an object of class <code>"abe"</code> , an object returned by a call to <code>[abe.resampling()]</code>
conf.level	the confidence level, defaults to 0.95, see <code>'details'</code>
pval	significance level to be used to determine a significant deviation from the expected pairwise inclusion frequency under independence.
alpha	the alpha value for which the output is to be printed. If <code>'NULL'</code> , the output is printed for all alpha values.
tau	the tau value for which the output is to be printed. If <code>'NULL'</code> , the output is printed for all tau values.
models.n	controls the number of models printed for <code>'model.rel.frequencies'</code> . See details.
...	additional arguments affecting the summary produced.

**Details**

Parameter `'conf.level'` defines the lower and upper quantile of the bootstrapped/resampled distribution such that equal proportion of values are smaller and larger than the lower and the upper quantile, respectively.

The `'models.n'` parameter controls the number of models printed in `'model.rel.frequencies'`. One option is to directly specify the number of models to return (i.e. an integer larger than 1). Alternatively, if `'models.n'` is set to a number less than (or equal to) 1, the number of models returned is such that the cumulative frequency attains that value. By default (`'models.n = NULL'`), the top 20 models or all models up to a cumulative frequency of 0.8, whichever is shorter, are returned. The selected model is marked with an asterisk. If it is not among the printed models, it is added as the last model.

**Value**

a list with the following elements:

`'var.rel.frequencies'`: inclusion relative frequencies for all variables from the initial model; if using `'type.resampling="Wallisch2021"'` in a call to `[abe.resampling()]` these results are based on subsampling with sampling proportion equal to 0.5, otherwise by using the method as specified by `'type.sampling'`



‘model.rel.frequencies’: relative frequencies of the final models; if using ‘type.resampling="Wallisch2021"’ in a call to [abe.resampling()] these results are based on subsampling with sampling proportion equal to 0.5, otherwise by using the method as specified by ‘type.sampling’

‘var.coefs’: coefficient estimates and standard errors from the global and the selected model and medians, means, percentiles and standard deviations for the resampled estimates for each variable from the initial model; if using ‘type.resampling="Wallisch2021"’ in a call to [abe.resampling()] these results are based on bootstrap, otherwise by using the method as specified by ‘type.sampling’

‘pair.rel.frequencies’: pairwise selection frequencies (in percent) for all pairs of variables. The significance of the deviation from the expected pairwise inclusion under independence is tested using a chi-squared test. If using ‘type.resampling="Wallisch2021"’ in a call to [abe.resampling()] these results are based on subsampling with sampling proportion equal to 0.5, otherwise by using the method as specified by ‘type.sampling’

### Author(s)

Rok Blagus, <rok.blagus@mf.uni-lj.si>  
 Sladana Babic  
 Daniela Dunkler  
 Gregor Steiner

### See Also

[abe.resampling](#), [print.abe](#), [plot.abe](#), [pie.abe](#)

### Examples

```
set.seed(1)
n=100
x1<-runif(n)
x2<-runif(n)
x3<-runif(n)
y<--5+5*x1+5*x2+ rnorm(n,sd=5)
dd<-data.frame(y=y,x1=x1,x2=x2,x3=x3)
fit<-lm(y~x1+x2+x3,x=TRUE,y=TRUE,data=dd)

fit.resample<-abe.resampling(fit,data=dd,include="x1",active="x2",
tau=c(0.05,0.1),exact=TRUE,
criterion="alpha",alpha=c(0.2,0.05),type.test="Chisq",
num.resamples=50,type.resampling="Wallisch2021")

summary(fit.resample)
```

# Index

abe, [2](#), [8](#)  
abe.resampling, [4](#), [5](#), [11](#), [12](#), [15](#), [17](#)  
coxph, [4](#)  
drop1, [3](#), [6](#)  
glm, [4](#)  
lm, [4](#)  
pie.abe, [8](#), [10](#), [12](#), [15](#), [17](#)  
plot.abe, [8](#), [11](#), [11](#), [15](#), [17](#)  
print.abe, [8](#), [13](#), [17](#)  
summary.abe, [8](#), [11](#), [12](#), [15](#), [15](#)