

Package ‘bootStepAIC’

February 2, 2026

Title Bootstrap stepAIC

Version 1.4-0

Date 2026-02-02

Maintainer Dimitris Rizopoulos <d.rizopoulos@erasmusmc.nl>

Description Model selection by bootstrapping the stepAIC() procedure.

Depends R(>= 4.0.0), MASS

LazyLoad yes

License GPL (>= 2)

NeedsCompilation no

Author Dimitris Rizopoulos [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-9397-0900>>)

Repository CRAN

Date/Publication 2026-02-02 10:40:02 UTC

Contents

boot.stepAIC 1

Index 4

boot.stepAIC	<i>Bootstraps the Stepwise Algorithm of stepAIC() for Choosing a Model by AIC</i>
--------------	---

Description

Implements a Bootstrap procedure to investigate the variability of model selection under the stepAIC() stepwise algorithm of package MASS.

Usage

```
boot.stepAIC(object, data, B = 100, alpha = 0.05, direction = "backward",  
            k = 2, verbose = FALSE, seed = 1L, ...)
```

Arguments

object	an object representing a model of an appropriate class; currently, "lm", "aov", "glm", "negbin", "polr", "survreg", and "coxph" objects are supported.
data	a <code>data.frame</code> or a <code>matrix</code> that contains the response variable and covariates.
B	the number of Bootstrap samples.
alpha	the significance level.
direction	the <code>direction</code> argument of <code>stepAIC()</code> .
k	the <code>k</code> argument of <code>stepAIC()</code> .
verbose	logical; if <code>TRUE</code> information about the evolution of the procedure is printed in the screen.
seed	numeric scalar denoting the seed used to create the Bootstrap samples.
...	extra arguments to <code>stepAIC()</code> , e.g., <code>scope</code> .

Details

The following procedure is replicated B times:

Step 1: Simulate a new data-set taking a sample with replacement from the rows of `data`.

Step 2: Refit the model using the data-set from Step 1.

Step 3: For the refitted model of Step 2 run the `stepAIC()` algorithm.

Summarize the results by counting how many times (out of the B data-sets) each variable was selected, how many times the estimate of the regression coefficient of each variable (out of the times it was selected) it was statistically significant in significance level `alpha`, and how many times the estimate of the regression coefficient of each variable (out of the times it was selected) changed signs (see also Austin and Tu, 2004).

Value

An object of class `BootStep` with components

Covariates	a numeric matrix containing the percentage of times each variable was selected.
Sign	a numeric matrix containing the percentage of times the regression coefficient of each variable had sign + and -.
Significance	a numeric matrix containing the percentage of times the regression coefficient of each variable was significant under the <code>alpha</code> significance level.
OrigModel	a copy of <code>object</code> .
OrigStepAIC	the result of applying <code>stepAIC()</code> in <code>object</code> .
direction	a copy of the <code>direction</code> argument.
k	a copy of the <code>k</code> argument.
BootStepAIC	a list of length B containing the results of <code>stepAIC()</code> for each Bootstrap data-set.

Author(s)

Dimitris Rizopoulos <d.rizopoulos@erasmusmc.nl>

References

Austin, P. and Tu, J. (2004). Bootstrap methods for developing predictive models, *The American Statistician*, **58**, 131–137.

Venables, W. N. and Ripley, B. D. (2002). Modern Applied Statistics with S, 4th ed. Springer, New York.

See Also

stepAIC in package MASS

Examples

```
## lm() Example ##
n <- 350
x1 <- runif(n, -4, 4)
x2 <- runif(n, -4, 4)
x3 <- runif(n, -4, 4)
x4 <- runif(n, -4, 4)
x5 <- runif(n, -4, 4)
x6 <- runif(n, -4, 4)
x7 <- factor(sample(letters[1:3], n, rep = TRUE))
y <- 5 + 3 * x1 + 2 * x2 - 1.5 * x3 - 0.8 * x4 + rnorm(n, sd = 2.5)
data <- data.frame(y, x1, x2, x3, x4, x5, x6, x7)
rm(n, x1, x2, x3, x4, x5, x6, x7, y)

lmFit <- lm(y ~ (. - x7) * x7, data = data)
boot.stepAIC(lmFit, data)

#####
## glm() Example ##
n <- 200
x1 <- runif(n, -3, 3)
x2 <- runif(n, -3, 3)
x3 <- runif(n, -3, 3)
x4 <- runif(n, -3, 3)
x5 <- factor(sample(letters[1:2], n, rep = TRUE))
eta <- 0.1 + 1.6 * x1 - 2.5 * as.numeric(as.character(x5) == levels(x5)[1])
y1 <- rbinom(n, 1, plogis(eta))
y2 <- rbinom(n, 1, 0.6)
data <- data.frame(y1, y2, x1, x2, x3, x4, x5)
rm(n, x1, x2, x3, x4, x5, eta, y1, y2)

glmFit1 <- glm(y1 ~ x1 + x2 + x3 + x4 + x5, family = binomial(), data = data)
glmFit2 <- glm(y2 ~ x1 + x2 + x3 + x4 + x5, family = binomial(), data = data)

boot.stepAIC(glmFit1, data, B = 50)
boot.stepAIC(glmFit2, data, B = 50)
```

Index

* **regression**

boot.stepAIC, [1](#)

boot.stepAIC, [1](#)