Package 'selcorr'

July 23, 2025

Title Post-Selection Inference for Generalized Linear Models

Version 1.0

Description Calculates (unconditional) post-selection confidence intervals and p-values for the coefficients of (generalized) linear models.

License GPL-3

Imports MASS, methods

Encoding UTF-8

RoxygenNote 7.1.2

NeedsCompilation no

Author Marco Cattaneo [aut, cre] (ORCID: <https://orcid.org/0000-0002-6610-8431>)

Maintainer Marco Cattaneo <marco.cattaneo@unibas.ch>

Repository CRAN

Date/Publication 2021-12-21 20:10:02 UTC

Contents

`selcorr` *Post-Selection Inference for Generalized Linear Models*

Description

`selcorr` returns (unconditional) post-selection confidence intervals and p-values for the coefficients of (generalized) linear models.

Usage

```
selcorr(
  object,
  fixed.vars = NULL,
  further.vars = NULL,
  boot.repl = 0,
  k = 2,
  conf.level = 0.95,
  quiet = FALSE
)
```

Arguments

`object`	an object representing a model of an appropriate class. This is used as the initial model in a (bidirectional) stepwise model selection.
`fixed.vars`	the names of all independent variables that must be included in the selected model. The default is none.
`further.vars`	the names of all independent variables that can be included in the selected model, but are not part of `object`. The default is none.
`boot.repl`	a number or list of bootstrap replicates. The default is no bootstrapping. See Details and Examples for clarification.
`k`	the multiple of the number of degrees of freedom used as penalty in the model selection. The default `k = 2` corresponds to the AIC.
`conf.level`	the level of the confidence intervals.
`quiet`	if `TRUE`, then `selcorr` does not generate an output.

Details

When `boot.repl = 0`, an approximate asymptotic distribution of the test statistic is used to calculate p-values and calibrate the profile-likelihood confidence intervals. This approach is faster, but p-values and confidence intervals can be more precisely calibrated by parametrically bootstrapping the test statistic (with `boot.repl` the number of replicates). Parallel computing can be used to speed up the bootstrapping: see Examples.

Value

the selected model is returned, without correction for model-selection, but with up to two additional components. There is an `output` component corresponding to the post-selection inference, which is also printed unless `quiet = TRUE`. When `boot.repl` is not `0`, there is also a `boot.repl` component corresponding to the bootstrap replicates.

Examples

```
## linear regression:
selcorr(lm(Fertility ~ ., swiss))

## logistic regression:
```

```
swiss.lr = within(swiss, Fertility <- (Fertility > 70))
selcorr(glm(Fertility ~ ., binomial, swiss.lr))

## parallel bootstrapping:
## Not run:
library(future.apply)
plan(multisession)
boot.repl = future_replicate(8, selcorr(lm(Fertility ~ ., swiss), boot.repl = 1000,
                                        quiet = TRUE)$boot.repl, simplify = FALSE)
plan(sequential)
selcorr(lm(Fertility ~ ., swiss), boot.repl = do.call("rbind", boot.repl))
## End(Not run)
```

Index