Package ‘boot.pval’

December 6, 2022

Title Bootstrap p-Values

Version 0.4.1

Description Computation of bootstrap p-values through inversion of confidence intervals, including convenience functions for regression models.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

RdMacros Rdpack

Imports boot, Rdpack, car, stats, lme4, survival

NeedsCompilation no

Author Måns Thulin [aut, cre]

Maintainer Måns Thulin <`mans@statistikkonsult.com`>

Repository CRAN

Date/Publication 2022-12-06 13:30:06 UTC

R topics documented:

`boot.pval` *Compute Bootstrap p-values*

Description

Compute bootstrap p-values through confidence interval inversion, as described in Hall (1992) and Thulin (2021).

Usage

```
boot.pval(boot_res, type = "perc", theta_null = 0, pval_precision = NULL, ...)
```

Arguments

`boot_res`	An object of class "boot" containing the output of a bootstrap calculation.
`type`	A vector of character strings representing the type of interval to base the test on. The value should be one of "norm", "basic", "stud", "perc" (the default), and "bca".
`theta_null`	The value of the parameter under the null hypothesis.
`pval_precision`	The desired precision for the p-value. The default is 1/R, where R is the number of bootstrap samples in `boot_res`.
`...`	Additional arguments passed to `boot.ci`.

Details

p-values can be computed by inverting the corresponding confidence intervals, as described in Section 12.2 of Thulin (2021) and Section 3.12 in Hall (1992). This function computes p-values in this way from "boot" objects. The approach relies on the fact that:

- the p-value of the two-sided test for the parameter theta is the smallest alpha such that theta is not contained in the corresponding 1-alpha confidence interval,
- for a test of the parameter theta with significance level alpha, the set of values of theta that aren't rejected by the two-sided test (when used as the null hypothesis) is a 1-alpha confidence interval for theta.

Value

A bootstrap p-value.

References

Hall P (1992). *The Bootstrap and Edgeworth Expansion*. Springer, New York. ISBN 9781461243847.

Thulin M (2021). *Modern Statistics with R*. Eos Chasma Press, Uppsala. ISBN 9789152701515, https://www.modernstatisticswithr.com/.

Examples

```
# Hypothesis test for the city data
# H0: ratio = 1.4
library(boot)
ratio <- function(d, w) sum(d$x * w)/sum(d$u * w)
city.boot <- boot(city, ratio, R = 99, stype = "w", sim = "ordinary")
boot.pval(city.boot, theta_null = 1.4)

# Studentized test for the two sample difference of means problem
# using the final two series of the gravity data.
diff.means <- function(d, f)
```

```
{
  n <- nrow(d)
  gp1 <- 1:table(as.numeric(d$series))[1]
  m1 <- sum(d[gp1,1] * f[gp1])/sum(f[gp1])
  m2 <- sum(d[-gp1,1] * f[-gp1])/sum(f[-gp1])
  ss1 <- sum(d[gp1,1]^2 * f[gp1]) - (m1 *  m1 * sum(f[gp1]))
  ss2 <- sum(d[-gp1,1]^2 * f[-gp1]) - (m2 *  m2 * sum(f[-gp1]))
  c(m1 - m2, (ss1 + ss2)/(sum(f) - 2))
}
grav1 <- gravity[as.numeric(gravity[,2]) >= 7, ]
grav1.boot <- boot(grav1, diff.means, R = 99, stype = "f",
                   strata = grav1[ ,2])
boot.pval(grav1.boot, type = "stud", theta_null = 0)
```

`boot_summary` *Summarising Regression Models Using the Bootstrap*

Description

Summaries for regression models, including "lm", "glm", "glm.nb", nls", "rlm", and "merMod" ("lmer", "glmer") objects, using the bootstrap for p-values and confidence intervals.

Usage

```
boot_summary(
  model,
  type = "perc",
  method = NULL,
  conf.level = 0.95,
  R = 999,
  coef = "raw",
  pval_precision = NULL,
  adjust.method = "none",
  ...
)
```

Arguments

`model`	An object fitted using e.g. "lm", "glm", "glm.nb", "nls", "rlm", "lmer", or "glmer".
`type`	A vector of character strings representing the type of interval to base the test on. The value should be one of "norm", "basic", "stud", "perc" (the default), and "bca". "stud" and "bca" are not available for "lmer" and "glmer" models.
`method`	The method used for bootstrapping. For "lm" and "nls" objects use either "residual" (for resampling of scaled and centred residuals, the default) or "case" (for case resampling). For "glm" objects, use "case" (the default). For "merMod" objects (mixed models) use either "parametric" (the default) or "semiparametric".

`conf.level`	The confidence level for the confidence intervals. The default is 0.95.
`R`	The number of bootstrap replicates. The default is 999.
`coef`	A string specifying whether to use exponentiated coefficients in the summary table. Either "exp" (for exponentiated coefficients, i.e. odds ratios in the case of a logistic regression model) or "raw" (for coefficients on their original scale). The default is "raw", which is recommended for linear models.
`pval_precision`	The desired precision for the p-value. The default is 1/R.
`adjust.method`	Adjustment of p-values for multiple comparisons using `p.adjust`. The default is "none", in which case the p-values aren't adjusted. The other options are "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", and "fdr"; see `?p.adjust` for details on these methods.
`...`	Additional arguments passed to `Boot` or `bootMer`, such as `parallel` for parallel computations. See `?car::Boot` and `?lme4::bootMer` for details.

Details

p-values can be computed by inverting the corresponding confidence intervals, as described in Section 12.2 of Thulin (2021) and Section 3.12 in Hall (1992). This function computes p-values for coefficients of regression models in this way. The approach relies on the fact that:

- the p-value of the two-sided test for the parameter theta is the smallest alpha such that theta is not contained in the corresponding 1-alpha confidence interval,
- for a test of the parameter theta with significance level alpha, the set of values of theta that aren't rejected by the two-sided test (when used as the null hypothesis) is a 1-alpha confidence interval for theta.

The function can be used with "lm", "glm", "glm.nb", "nls", "rlm", and "merMod" ("lmer", "glmer") objects. In addition, it should work for any regression model such that: `residuals(object, type="pearson")` returns Pearson residuals; `fitted(object)` returns fitted values; `hatvalues(object)` returns the leverages, or perhaps the value 1 which will effectively ignore setting the hatvalues. In addition, the `data` argument should contain no missing values among the columns actually used in fitting the model.

Value

A data frame containing coefficient estimates, bootstrap confidence intervals, and bootstrap p-values.

References

Hall P (1992). *The Bootstrap and Edgeworth Expansion*. Springer, New York. ISBN 9781461243847.

Thulin M (2021). *Modern Statistics with R*. Eos Chasma Press, Uppsala. ISBN 9789152701515, `https://www.modernstatisticswithr.com/`.

Examples

```
# Bootstrap summary of a linear model for mtcars:
model <- lm(mpg ~ hp + vs, data = mtcars)
boot_summary(model, R = 99)
# (Values for R greater than 99 are recommended for most applications.)

# Adjust p-values for multiplicity using Holm's method:
boot_summary(model, R = 99, adjust.method = "holm")
```

`censboot_summary` *Summarising Survival Regression Models Using the Bootstrap*

Description

Summaries for "coxph" and "survreg" objects, using the bootstrap for p-values and confidence intervals.

Usage

```
censboot_summary(
  model,
  type = "perc",
  sim = "ordinary",
  strata = NULL,
  coef = "exp",
  conf.level = 0.95,
  R = 999,
  pval_precision = NULL,
  adjust.method = "none",
  ...
)
```

Arguments

`model`	An object fitted using "coxph" or "survreg".
`type`	A vector of character strings representing the type of interval to base the test on. The value should be one of "norm", "basic", "stud", "perc" (the default), and "bca".
`sim`	The method used for bootstrapping. See ?`boot::censboot` for details. Currently only "ordinary" (case resampling) is supported.
`strata`	The strata used in the calls to `survfit`. It can be a vector or a matrix with 2 columns. If it is a vector then it is assumed to be the strata for the survival distribution, and the censoring distribution is assumed to be the same for all observations. If it is a matrix then the first column is the strata for the survival distribution and the second is the strata for the censoring distribution. When `sim = "ordinary"`, only one set of strata is used to stratify the observations. This is taken to be the first column of `strata` when it is a matrix.

`coef`	A string specifying whether to use exponentiated coefficients in the summary table. Either "exp" (for exponentiated coefficients, i.e. hazard ratios in the case of a Cox PH model) or "raw" (for coefficients on their original scale). The default is "exp".
`conf.level`	The confidence level for the confidence intervals. The default is 0.95.
`R`	The number of bootstrap replicates. The default is 999.
`pval_precision`	The desired precision for the p-value. The default is 1/R.
`adjust.method`	Adjustment of p-values for multiple comparisons using `p.adjust`. The default is "none", in which case the p-values aren't adjusted. The other options are "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", and "fdr"; see `?p.adjust` for details on these methods.
`...`	Additional arguments passed to `censboot`, such as `parallel` for parallel computations. See `?boot::censboot` for details.

Details

p-values can be computed by inverting the corresponding confidence intervals, as described in Section 12.2 of Thulin (2021) and Section 3.12 in Hall (1992). This function computes p-values in this way from "coxph" or "survreg" objects. The approach relies on the fact that:

- the p-value of the two-sided test for the parameter theta is the smallest alpha such that theta is not contained in the corresponding 1-alpha confidence interval,
- for a test of the parameter theta with significance level alpha, the set of values of theta that aren't rejected by the two-sided test (when used as the null hypothesis) is a 1-alpha confidence interval for theta.

Value

A data frame containing coefficient estimates, bootstrap confidence intervals, and bootstrap p-values.

References

Hall P (1992). *The Bootstrap and Edgeworth Expansion*. Springer, New York. ISBN 9781461243847.

Thulin M (2021). *Modern Statistics with R*. Eos Chasma Press, Uppsala. ISBN 9789152701515, https://www.modernstatisticswithr.com/.

Examples

```
library(survival)
# Weibull AFT model:
# Note that model = TRUE is required for use with censboot_summary:
model <- survreg(formula = Surv(time, status) ~ age + sex, data = lung,
                 dist = "weibull", model = TRUE)
censboot_summary(model, R = 99)
# (Values for R greater than 99 are recommended for most applications.)

# Cox PH model:
```

```
model <- coxph(formula = Surv(time, status) ~ age + sex, data = lung,
               model = TRUE)
# Table with hazard ratios:
censboot_summary(model, R = 99)
censboot_summary(model, coef = "raw", R = 99)
```

Index