Package ‘hcinfer’

August 4, 2026

Title Heteroskedasticity-Consistent Inference for Linear Models

Version 0.2.0

Description Computes heteroskedasticity-consistent covariance matrix estimators for ordinary least squares regression models. The published HC0 through HC5m estimators implemented in the package follow White (1980) <doi:10.2307/1912934>, Hinkley (1977) <doi:10.1080/00401706.1977.10489550>, MacKinnon and White (1985) <doi:10.1016/0304-4076(85)90158-7>, Cribari-Neto (2004) <doi:10.1016/S0167-9473(02)00366-3>, Cribari-Neto and da Silva (2011) <doi:10.1007/s10182-010-0141-2>, Cribari-Neto et al. (2007) <doi:10.1080/03610920601126589>, and Li et al. (2016) <doi:10.1080/00949655.2016.1198906>. The package also includes HCbeta, a new estimator proposed by the package authors. It provides normal Wald tests, confidence intervals, diagnostics, and S3 output for applied inference.

URL `https://prdm0.github.io/hcinfer/`, `https://github.com/prdm0/hcinfer`

BugReports `https://github.com/prdm0/hcinfer/issues`

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports cli, ggplot2, purrr, rlang, tibble

Suggests carrier, dplyr, knitr, mirai, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/Needs/website pkgdown

LazyData true

Config/roxygen2/version 8.0.0

RoxygenNote 8.0.0

NeedsCompilation no

Author Pedro Rafael D. Marinho [aut, cre] (ORCID: <https://orcid.org/0000-0003-1591-8300>), Francisco Cribari-Neto [aut] (ORCID: <https://orcid.org/0000-0002-5909-6698>), Marina Oliveira Cunha [aut] (ORCID: <https://orcid.org/0009-0007-4938-3881>)

Maintainer Pedro Rafael D. Marinho <pedro.rafael.marinho@gmail.com>

Repository CRAN

Date/Publication 2026-08-04 18:30:02 UTC

Contents

`boot_pairs` *Pairs bootstrap standard errors and confidence intervals*

Description

Computes pairs (case) bootstrap standard errors and confidence intervals for the coefficients of an ordinary least squares model fitted with `stats::lm()`. The pairs bootstrap resamples the observations (y_t, x_t) with replacement, refits OLS on each resample, and summarizes the resulting sampling distribution of $\hat{\beta}$. It makes no assumption about the form of the error variance, so it is a useful empirical reference for the analytic heteroskedasticity-consistent standard errors produced by `hcinfer()` and `vcov_hc()`.

Usage

```
boot_pairs(
  object,
  B = 1000L,
  level = 0.95,
  ci_type = c("percentile", "basic", "normal"),
  cores = 1L,
  seed = NULL
)

## S3 method for class 'hcinfer_boot'
print(x, ...)
```

Arguments

`object`	An ordinary least squares model fitted by `stats::lm()`. Weighted fits are not supported.
`B`	Number of bootstrap replicates. A positive integer; defaults to `1000`.
`level`	Confidence level for the intervals, strictly between 0 and 1. Defaults to `0.95`.
`ci_type`	Interval type: `"percentile"` (default), `"basic"`, or `"normal"`. See Details.
`cores`	Number of worker processes. A single number greater than or equal to 1; non-integer values are rounded to the nearest integer. The default `1` runs the replicate fits sequentially. Any value that rounds to `2` or more runs them in parallel with `purrr::in_parallel()` and the **mirai** backend, which requires the **mirai** and **carrier** packages. Parallelism only speeds up the computation and never changes the numeric result.
`seed`	Optional single number used to seed the resampling for reproducibility. When supplied, the result is deterministic and independent of the number of `cores`. Defaults to `NULL`.
`x`	An object returned by `boot_pairs()`.
`...`	Unused.

Details

For each of `B` bootstrap replicates, a sample of `n` row indices is drawn with replacement from `1, ..., n`, and OLS is refitted on the resampled rows (y_i, x_i). Writing $\hat\beta^*_{(r)}$ for the estimate on replicate `r`, the bootstrap standard error of coefficient `j` is the sample standard deviation of $\hat\beta^*_{1,j}, \ldots, \hat\beta^*_{B,j}$.

Three interval types are available through `ci_type`. Let q_α denote the empirical α quantile of the bootstrap replicates for a coefficient, $\hat\beta_j$ the original estimate, s^*_j the bootstrap standard error, and $\alpha = 1 -$ `level`. The `"percentile"` interval is $[q_{\alpha/2}, q_{1-\alpha/2}]$. The `"basic"` (reverse percentile) interval is $[2\hat\beta_j - q_{1-\alpha/2},\ 2\hat\beta_j - q_{\alpha/2}]$. The `"normal"` interval is $\hat\beta_j \pm z_{1-\alpha/2}\, s^*_j$, with z the standard normal quantile.

Reproducibility. When `seed` is supplied, all resampling indices are drawn once, sequentially, in the main process under that seed, and the per-replicate fit is deterministic. The results are therefore identical whether the run is sequential or parallel and regardless of the number of cores. The caller's random number generator state is saved and restored, so calling `boot_pairs()` does not disturb a surrounding random stream. When `seed` is `NULL`, the current RNG state is used and results are not reproducible.

Parallelism. The default `cores = 1` fits the replicates sequentially. When `cores` rounds to `2` or more, the deterministic per-replicate fits are distributed with `purrr::in_parallel()`, which uses the **mirai** package as its backend. `boot_pairs()` starts `cores` daemons for the duration of the call and shuts them down on exit; do not call it while relying on externally configured `mirai` daemons. Parallelism only speeds up the computation: it never changes the numeric result. It is worthwhile mainly for large `B` or large `n`; for small problems the setup overhead can dominate.

Rank-deficient resamples. A resample can be rank deficient (for example when a resample omits the observations that identify a coefficient). Such replicates are dropped, a warning reports how many were dropped, and the summaries use the remaining replicates. The call errors if fewer than two valid replicates remain.

Value

An object of class `hcinfer_boot`: a list with the original OLS `coefficients`, bootstrap `std_error`, `bias`, interval endpoints `conf_low` and `conf_high`, the settings (`level`, `ci_type`, `B`, `B_effective`, `n_failed`, `cores`, `seed`), the full `replicates` matrix (`B` rows by `p` columns), and a tidy `table` tibble with columns `term`, `estimate`, `bias`, `std_error`, `conf_low`, and `conf_high`. Use `coef()`, `vcov()`, and `confint()` to extract components.

References

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap Methods and their Application*. Cambridge University Press. doi:10.1017/CBO9780511802843

Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman and Hall. doi:10.1201/9780429246593

See Also

`hcinfer()`, `vcov_hc()`

Examples

```
schools <- PublicSchools |>
  dplyr::mutate(
    income_scaled = income / 10000,
    income_scaled_sq = income_scaled^2
  )
fit <- lm(expenditure ~ income_scaled + income_scaled_sq, data = schools)

# 1. Fit, inspect, and visualize a reproducible pairs bootstrap.
boot <- boot_pairs(fit, B = 1000, seed = 123)
boot
confint(boot)
plot(boot)

# 2. Use the bootstrap as an empirical reference for the analytic HC standard
```

```
#    errors, side by side in one table.
data.frame(
  term = boot$table$term,
  ols = sqrt(diag(vcov(fit))),
  bootstrap = boot$table$std_error,
  hcbeta = sqrt(diag(vcov(hcinfer(fit, type = "hcbeta")))),
  hc3 = sqrt(diag(vcov(hcinfer(fit, type = "hc3"))))
)

# 3. Recompute intervals at a new level and type from the same replicates,
#    without rerunning the bootstrap.
confint(boot, level = 0.99, type = "basic")
confint(boot, parm = "income_scaled_sq", level = 0.90)

# 4. Larger, parallel run on two cores. Requires the mirai and carrier
#    packages; the numeric result matches a sequential run with the same seed.

if (requireNamespace("mirai", quietly = TRUE) &&
    requireNamespace("carrier", quietly = TRUE)) {
  boot_par <- boot_pairs(fit, B = 4000, ci_type = "basic", cores = 2, seed = 42)
  boot_par$table
}
```

`coef.hcinfer` *Extract model coefficients from an hcinfer object*

Description

Extracts the OLS coefficients stored in an `hcinfer()` result.

Usage

```
## S3 method for class 'hcinfer'
coef(object, ...)
```

Arguments

`object`	An object returned by `hcinfer()`.
`...`	Unused.

Value

A named numeric vector of OLS coefficients.

confint.hcinfer — *Confidence intervals for hcinfer objects*

Description

Extracts normal Wald confidence intervals from an hcinfer() result. If the requested level differs from the level used to create the object, only the normal critical value and interval endpoints are recomputed.

Usage

```
## S3 method for class 'hcinfer'
confint(object, parm, level = object$confidence_level, ...)
```

Arguments

`object`	An object returned by hcinfer().
`parm`	Optional coefficient names or positions.
`level`	Confidence level.
`...`	Unused.

Value

A tibble with columns `term`, `conf_low`, `conf_high`, and `level`.

Crime2009 — *State crime rates and socioeconomic indicators, 2009*

Description

Violent-crime and murder rates together with socioeconomic indicators for the 50 U.S. states and the District of Columbia in 2009. The data are useful for illustrating heteroskedasticity-consistent inference in a cross-sectional design with influential observations.

Usage

```
Crime2009
```

Format

A tibble with 51 rows and 8 variables:

state Name of one of the 50 U.S. states or the District of Columbia.

violent Violent-crime rate per 100,000 population.

murder Murder rate per 100,000 population.

hs_grad Percentage of the population that graduated from high school or higher.

poverty Percentage of the population living below the poverty line.

single Percentage of households headed by a single parent.

white Percentage of the population that is white.

urban Percentage of the population living in urban areas.

Source

French, J. P. (2023). *api2lm: Functions and Data Sets for the Book 'A Progressive Introduction to Linear Models'*. R package version 0.2. doi:10.32614/CRAN.package.api2lm. The same data are distributed as the `statecrime` dataset in the Python `statsmodels` package (Seabold and Perktold, 2010, `https://www.statsmodels.org/`); the underlying figures come from the *Statistical Abstract of the United States* (2009) and are in the public domain.

Examples

```
data(Crime2009)
Crime2009[Crime2009$state == "Alabama", ]

fit <- lm(murder ~ hs_grad + poverty + single, data = Crime2009)
hcinfer(fit, type = "hcbeta")
```

`hcinfer` *Heteroskedasticity-consistent Wald inference*

Description

Computes normal Wald tests and confidence intervals for an ordinary least squares model using a heteroskedasticity-consistent covariance estimator.

Usage

```
hcinfer(object, type = "hcbeta", alpha = 0.05, null = 0, ...)
```

Arguments

`object`	An ordinary least squares model fitted by `stats::lm()`.
`type`	A character string specifying the HC estimator. The default is `"hcbeta"`.
`alpha`	Significance level. The confidence level is `1 - alpha`.
`null`	Null values for the coefficient tests. Use a scalar to test all coefficients against the same value, or a numeric vector with one value per coefficient.
`...`	Method-specific constants passed to `vcov_hc()`. For HCbeta, `a_max` and `b_max` default to 10000, may be set independently, and must each be finite and lie in `[50, 25000]`. See `vcov_hc()` for all other method-specific defaults and parameter domains.

Details

For each coefficient, hcinfer tests

$$H_0 : \beta_j = \beta_j^{(0)}$$

against a two-sided alternative using the statistic

$$z_j = \frac{\hat{\beta}_j - \beta_j^{(0)}}{\sqrt{[\widehat{\Psi}_{HC}]_{jj}}}.$$

The reference distribution is the standard normal distribution. Confidence intervals are Wald intervals obtained by direct inversion of the test,

$$\hat{\beta}_j \pm z_{1-\alpha/2}\sqrt{[\widehat{\Psi}_{HC}]_{jj}}.$$

Bootstrap intervals and Student t quantiles are not used.

Value

An object of class `hcinfer` containing the fitted HC covariance estimator, coefficient tests, p-values, confidence intervals, diagnostics, and method parameters.

References

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4), 817-838. doi:10.2307/1912934

Hinkley, D. V. (1977). Jackknifing in unbalanced situations. *Technometrics*, 19(3), 285-292. doi:10.1080/00401706.1977.10489550

MacKinnon, J. G. and White, H. (1985). Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, 29(3), 305-325. doi:10.1016/03044076(85)901587

Davidson, R. and MacKinnon, J. G. (1993). *Estimation and Inference in Econometrics*. Oxford University Press.

Cribari-Neto, F. (2004). Asymptotic inference under heteroskedasticity of unknown form. *Computational Statistics and Data Analysis*, 45(2), 215-233. doi:10.1016/S01679473(02)003663

Cribari-Neto, F. and da Silva, W. B. (2011). A new heteroskedasticity consistent covariance matrix estimator for the linear regression model. *AStA Advances in Statistical Analysis*, 95(2), 129-146. doi:10.1007/s1018201001412

Cribari-Neto, F., Souza, T. C., and Vasconcellos, K. L. P. (2007). Inference under heteroskedasticity and leveraged data. *Communications in Statistics - Theory and Methods*, 36(10), 1877-1888. doi:10.1080/03610920601126589

Li, S., Zhang, N., Zhang, X., and Wang, G. (2016). A new heteroskedasticity-consistent covariance matrix estimator and inference under heteroskedasticity. *Journal of Statistical Computation and Simulation*, 87(1), 198-210. doi:10.1080/00949655.2016.1198906

Examples

```
schools <- PublicSchools |>
  dplyr::mutate(
    income_scaled = income / 10000,
    income_scaled_sq = income_scaled^2
  )
fit <- lm(expenditure ~ income_scaled + income_scaled_sq, data = schools)
result <- hcinfer(fit, type = "hcbeta")
result
summary(result)
confint(result)

# Sensitivity analysis with nondefault HCbeta caps
hcinfer(fit, type = "hcbeta", a_max = 20000, b_max = 20000)
hcinfer(fit, type = "hc5", k = 0.7)
hcinfer(fit, type = "hc5m", k = 0.7, k1 = 1, k2 = 0, k3 = 1)
```

`hcinfer_boot-methods` *Extract components from a pairs bootstrap object*

Description

Extractors for objects returned by `boot_pairs()`. `coef()` returns the original OLS coefficients, `vcov()` returns the bootstrap covariance matrix (the sample covariance of the bootstrap replicates), and `confint()` returns bootstrap confidence intervals, optionally recomputed at a different `level` or `type` from the stored replicates.

Usage

```
## S3 method for class 'hcinfer_boot'
coef(object, ...)

## S3 method for class 'hcinfer_boot'
```

```
vcov(object, ...)

## S3 method for class 'hcinfer_boot'
confint(object, parm, level = object$level, type = object$ci_type, ...)
```

Arguments

object	An object returned by boot_pairs().
...	Unused.
parm	Optional coefficient names or integer positions.
level	Confidence level for confint(). Defaults to the level stored in object.
type	Interval type for confint(): "percentile", "basic", or "normal". Defaults to the type stored in object.

Value

coef() a named numeric vector; vcov() a numeric covariance matrix; confint() a tibble with columns term, conf_low, conf_high, and level.

hc_methods — *Available heteroskedasticity-consistent estimators*

Description

Returns the HC covariance estimators implemented by hcinfer.

Usage

```
hc_methods()
```

Value

A tibble with columns type, label, description, and default_arguments.

Examples

```
hc_methods()
```

`Hprice` *Boston-area home prices, 1990*

Description

Sale prices, assessed values, and physical characteristics of 88 homes sold in the Boston, Massachusetts area in 1990. The data are widely used to illustrate regression and heteroskedasticity-consistent inference.

Usage

```
Hprice
```

Format

A tibble with 88 rows and 10 variables:

price House price, in thousands of U.S. dollars.

assess Assessed value, in thousands of U.S. dollars.

bdrms Number of bedrooms.

lotsize Size of the lot, in square feet.

sqrft Size of the house, in square feet.

colonial Indicator equal to 1 if the home is of colonial style.

lprice Natural logarithm of `price`.

lassess Natural logarithm of `assess`.

llotsize Natural logarithm of `lotsize`.

lsqrft Natural logarithm of `sqrft`.

Source

Wooldridge, J. M. (2020). *Introductory Econometrics: A Modern Approach*, 7th ed. Cengage Learning, Boston, MA. The `hprice1` data are distributed with the `wooldridge` R package and were originally collected from the real estate pages of the *Boston Globe*.

Examples

```
data(Hprice)
head(Hprice)

fit <- lm(price ~ lotsize + bdrms + bdrms:sqrft, data = Hprice)
hcinfer(fit, type = "hcbeta")
```

plot.hcinfer *Plot robust confidence intervals*

Description

Plots normal Wald confidence intervals for an hcinfer() result. Each interval is color-coded by the test decision at the stored significance level: coefficients for which the null hypothesis is rejected are shown in red, and those for which it is not rejected are shown in blue. Formatted p-values are printed to the right of each interval for quick reading.

Usage

```
## S3 method for class 'hcinfer'
plot(x, parm, ...)
```

Arguments

x	An object returned by hcinfer().
parm	Optional coefficient names or integer positions. When supplied, only the selected coefficients are plotted. The selection follows the same rules as confint.hcinfer() and tests.hcinfer().
...	Unused. Passing named arguments raises an error.

Value

A ggplot2::ggplot() object.

See Also

hcinfer(), confint.hcinfer(), tests.hcinfer()

Examples

```
schools <- PublicSchools |>
  dplyr::mutate(
    income_scaled = income / 10000,
    income_scaled_sq = income_scaled^2
  )
fit <- lm(expenditure ~ income_scaled + income_scaled_sq, data = schools)
result <- hcinfer(fit)
plot(result)
plot(result, parm = "income_scaled_sq")
```

`plot.hcinfer_boot` *Plot pairs bootstrap confidence intervals*

Description

Plots the pairs bootstrap confidence intervals stored in a `boot_pairs()` object. Each coefficient is drawn as its ordinary least squares point estimate with a horizontal bootstrap interval, color-coded by whether the interval excludes zero (shown in red) or includes zero (shown in blue). A dashed vertical reference line is drawn at zero.

Usage

```
## S3 method for class 'hcinfer_boot'
plot(x, parm, ...)
```

Arguments

`x`	An object returned by `boot_pairs()`.
`parm`	Optional coefficient names or integer positions. When supplied, only the selected coefficients are plotted, following the same rules as `confint.hcinfer_boot()`.
`...`	Unused. Passing named arguments raises an error.

Value

A `ggplot2::ggplot()` object.

See Also

`boot_pairs()`, `plot.hcinfer()`

Examples

```
schools <- PublicSchools |>
  dplyr::mutate(
    income_scaled = income / 10000,
    income_scaled_sq = income_scaled^2
  )
fit <- lm(expenditure ~ income_scaled + income_scaled_sq, data = schools)
boot <- boot_pairs(fit, B = 1000, seed = 123)
plot(boot)
plot(boot, parm = "income_scaled_sq")
```

plot.hcinfer_vcov *Plot HC adjustment factors against leverages*

Description

Plots the HC adjustment factors g_t against the leverage values h_t stored in a vcov_hc() object. Points with $h_t > 3p/n$ are highlighted because this threshold is commonly used to flag high-leverage observations in the empirical examples from the HCbeta paper.

Usage

```
## S3 method for class 'hcinfer_vcov'
plot(x, label_top = 3, ...)
```

Arguments

x	An object returned by vcov_hc().
label_top	A nonnegative whole number. The observations with the largest adjustment factors are labeled. Use 0 to suppress labels.
...	Unused. Passing named arguments raises an error.

Value

A ggplot2::ggplot() object.

See Also

vcov_hc(), hcinfer(), plot.hcinfer()

Examples

```
schools <- PublicSchools |>
  dplyr::mutate(
    income_scaled = income / 10000,
    income_scaled_sq = income_scaled^2
  )
fit <- lm(expenditure ~ income_scaled + income_scaled_sq, data = schools)

cov <- vcov_hc(fit, type = "hcbeta")
plot(cov)
plot(vcov_hc(fit, type = "hc4"), label_top = 2)
```

`print.hcinfer` *Print hcinfer objects*

Description

Prints a compact overview of a heteroskedasticity-consistent inference object. Emoji markers are used when the current locale supports UTF-8 and `getOption("hcinfer.use_emoji", TRUE)` is true.

Usage

```
## S3 method for class 'hcinfer'
print(x, ...)
```

Arguments

`x`	An object returned by `hcinfer()`.
`...`	Unused.

Value

The input object, invisibly.

`print.hcinfer_vcov` *Print hcinfer covariance objects*

Description

Prints a compact overview of a heteroskedasticity-consistent covariance object. Emoji markers are used when the current locale supports UTF-8 and `getOption("hcinfer.use_emoji", TRUE)` is true.

Usage

```
## S3 method for class 'hcinfer_vcov'
print(x, ...)
```

Arguments

`x`	An object returned by `vcov_hc()`.
`...`	Unused.

Value

The input object, invisibly.

`PublicSchools` *Public school expenditure and income by U.S. jurisdiction*

Description

Public school expenditure and income data for the 50 U.S. states and the District of Columbia in 1979. The expenditure value for Wisconsin is missing in the source data, so the standard regression example uses 50 complete observations. The data are useful for illustrating heteroskedasticity-consistent inference because Alaska is a high-leverage observation in the quadratic public-schools model studied in the HCbeta paper.

Usage

```
PublicSchools
```

Format

A tibble with 51 rows and 3 variables:

state Name of one of the 50 U.S. states or the District of Columbia.

expenditure Per capita expenditure on public schools in 1979. This variable has one missing value.

income Per capita income in 1979.

Source

Greene, W. H. (1993). *Econometric Analysis*, 2nd ed. Macmillan Publishing Company, New York. Table 14.1, p. 385. The data were originally sourced from the U.S. Department of Commerce, *Statistical Abstract of the United States* (1979). The dataset is also available in the `sandwich` R package.

Examples

```
data(PublicSchools)
PublicSchools[PublicSchools$state == "Alaska", ]

schools <- PublicSchools |>
  dplyr::mutate(
    income_scaled = income / 10000,
    income_scaled_sq = income_scaled^2
  )
fit <- lm(expenditure ~ income_scaled + income_scaled_sq, data = schools)
hcinfer(fit, type = "hcbeta")
```

PublicSchools2 *Public school expenditure, income, and region by U.S. jurisdiction*

Description

Public school expenditure and per capita income for the 50 U.S. states and the District of Columbia. Income is measured for 2024, and expenditure is measured for 2025. The regional indicator uses the U.S. Census Bureau classification of the Southern United States.

Usage

```
PublicSchools2
```

Format

A tibble with 51 rows and 4 variables:

state Character. Name of one of the 50 U.S. states or the District of Columbia.

income Integer. Annual per capita personal income for 2024, in nominal U.S. dollars. It is calculated as total personal income for the jurisdiction divided by its population.

expenditure Integer. Annual expenditure per student enrolled in K-12 public schools for 2025, in U.S. dollars. It includes instructional salaries and expenses, school support, and administrative services.

south Integer. Indicator equal to 1 for Alabama, Arkansas, Delaware, the District of Columbia, Florida, Georgia, Kentucky, Louisiana, Maryland, Mississippi, North Carolina, Oklahoma, South Carolina, Tennessee, Texas, Virginia, and West Virginia, and 0 otherwise.

Source

World Population Review (2026), *Per Capita Income by State*, https://worldpopulationreview.com/state-rankings/per-capita-income-by-state. Accessed June 11, 2026. The supplied data dictionary also attributes the income measure to the U.S. Bureau of Economic Analysis.

World Population Review (2026), *Per Pupil Spending by State*, https://worldpopulationreview.com/state-rankings/per-pupil-spending-by-state. Accessed June 11, 2026.

U.S. Census Bureau, *Terms and Definitions: Census Regions and Divisions*, https://www.census.gov/programs-surveys/popest/guidance-geographies/terms-and-definitions.html.

Wikipedia, *Southern United States*, https://en.wikipedia.org/wiki/Southern_United_States. This was the geographic source recorded in the supplied data dictionary. Accessed June 11, 2026.

Examples

```
data(PublicSchools2)
PublicSchools2[PublicSchools2$state == "District of Columbia", ]
table(PublicSchools2$south)
```

summary.hcinfer — *Summarize heteroskedasticity-consistent inference*

Description

Builds a detailed summary for an `hcinfer()` result. The summary includes model metadata, HC method information, leverage diagnostics, robust weight diagnostics, and coefficient-by-coefficient normal Wald tests with p-values and confidence intervals. The print method adds formal test decisions to improve interpretation while preserving the numeric components of the object.

Usage

```
## S3 method for class 'hcinfer'
summary(object, ...)
```

Arguments

`object`	An object returned by `hcinfer()`.
`...`	Unused.

Value

An object of class `summary_hcinfer`.

summary.hcinfer_vcov — *Summarize heteroskedasticity-consistent covariance objects*

Description

Builds a detailed summary for an object returned by `vcov_hc()`.

Usage

```
## S3 method for class 'hcinfer_vcov'
summary(object, ...)
```

Arguments

`object`	An object returned by `vcov_hc()`.
`...`	Unused.

Value

An object of class `summary_hcinfer_vcov`.

`tests` *Extract coefficient test results*

Description

Extracts the normal Wald test results from an `hcinfer()` object. If the requested significance level differs from the one used to create the object, only the `reject` column is recomputed. The test statistics and p-values are not affected by `alpha` and are never recomputed.

Usage

```
tests(object, ...)

## S3 method for class 'hcinfer'
tests(object, parm, alpha = object$alpha, ...)
```

Arguments

`object`	An object returned by `hcinfer()`.
`...`	Unused. Passing named arguments raises an error.
`parm`	Optional coefficient names or integer positions to select a subset of coefficients. When omitted, all coefficients are returned.
`alpha`	Significance level used to compute the `reject` column. Must be strictly between 0 and 1. Defaults to the level stored in `object`. Changing `alpha` updates only the `reject` column; all other columns remain identical to the stored values.

Details

For each coefficient, the stored test is

$$H_0 : \beta_j = \beta_j^{(0)}$$

against a two-sided alternative. The test statistic is

$$z_j = \frac{\hat{\beta}_j - \beta_j^{(0)}}{\sqrt{[\widehat{\Psi}_{HC}]_{jj}}},$$

and the p-value is $2\,\Phi(-|z_j|)$, where Φ is the standard normal distribution function. The null value $\beta_j^{(0)}$ is the one stored in the object, set when `hcinfer()` was called.

To test against a different null value, rerun `hcinfer()` with the desired `null` argument.

Value

A tibble with one row per selected coefficient and the following columns:

`term` Coefficient name.

`estimate` OLS estimate $\hat{\beta}_j$.

`null_value` Null hypothesis value $\beta_j^{(0)}$.

`std_error` Robust standard error $\sqrt{[\widehat{\Psi}_{HC}]_{jj}}$.

`z_value` Normal Wald statistic z_j.

`p_value` Two-sided p-value $2\,\Phi(-|z_j|)$.

`alpha` Significance level used for the `reject` column.

`reject` Logical. `TRUE` when `p_value` < `alpha`.

References

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4), 817-838. doi:10.2307/1912934

Hinkley, D. V. (1977). Jackknifing in unbalanced situations. *Technometrics*, 19(3), 285-292. doi:10.1080/00401706.1977.10489550

MacKinnon, J. G. and White, H. (1985). Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, 29(3), 305-325. doi:10.1016/03044076(85)901587

Davidson, R. and MacKinnon, J. G. (1993). *Estimation and Inference in Econometrics*. Oxford University Press.

Cribari-Neto, F. (2004). Asymptotic inference under heteroskedasticity of unknown form. *Computational Statistics and Data Analysis*, 45(2), 215-233. doi:10.1016/S01679473(02)003663

Cribari-Neto, F. and da Silva, W. B. (2011). A new heteroskedasticity consistent covariance matrix estimator for the linear regression model. *AStA Advances in Statistical Analysis*, 95(2), 129-146. doi:10.1007/s1018201001412

Cribari-Neto, F., Souza, T. C., and Vasconcellos, K. L. P. (2007). Inference under heteroskedasticity and leveraged data. *Communications in Statistics - Theory and Methods*, 36(10), 1877-1888. doi:10.1080/03610920601126589

Li, S., Zhang, N., Zhang, X., and Wang, G. (2016). A new heteroskedasticity-consistent covariance matrix estimator and inference under heteroskedasticity. *Journal of Statistical Computation and Simulation*, 87(1), 198-210. doi:10.1080/00949655.2016.1198906

See Also

`hcinfer()`, `confint.hcinfer()`

Examples

```
schools <- PublicSchools |>
  dplyr::mutate(
    income_scaled = income / 10000,
    income_scaled_sq = income_scaled^2
  )
fit <- lm(expenditure ~ income_scaled + income_scaled_sq, data = schools)
result <- hcinfer(fit)

tests(result)
tests(result, parm = "income_scaled_sq")
tests(result, alpha = 0.10)
```

`vcov.hcinfer` *Extract robust covariance matrices*

Description

Extracts the heteroskedasticity-consistent covariance matrix stored in an hcinfer object. The matrix is returned directly and is not recomputed.

Usage

```
## S3 method for class 'hcinfer'
vcov(object, ...)

## S3 method for class 'hcinfer_vcov'
vcov(object, ...)
```

Arguments

`object`	An object returned by `hcinfer()` or `vcov_hc()`.
`...`	Unused.

Value

A numeric covariance matrix.

vcov_hc *Heteroskedasticity-consistent covariance estimator*

Description

Computes a heteroskedasticity-consistent covariance matrix estimator for an ordinary least squares model fitted with `stats::lm()`. The function returns a rich S3 object that stores the covariance matrix, HC weights, leverage values, method parameters, and model metadata.

Usage

```
vcov_hc(object, type = "hcbeta", ...)
```

Arguments

`object`	An ordinary least squares model fitted by `stats::lm()`.
`type`	A character string specifying the HC estimator. The default is `"hcbeta"`.
`...`	Method-specific constants. Unknown names are rejected. See Details for the accepted names, defaults, and parameter domains.

Details

For a linear model with design matrix X, OLS residuals $\hat{e}_t$, and HC weights g_t, the estimator is

$$\widehat{\Psi}_{HC} = (X'X)^{-1}X'\widehat{\Omega}X(X'X)^{-1},$$

where $\widehat{\Omega} = diag(\hat{e}_t^2 g_t)$. The supported estimators are `"hc0"`, `"hc1"`, `"hc2"`, `"hc3"`, `"hc4"`, `"hc4m"`, `"hc5"`, `"hc5m"`, and `"hcbeta"`.

Additional arguments in `...` are method-specific. The defaults are:

- `"hc0"`, `"hc1"`, `"hc2"`, `"hc3"`, `"hc4"`, and `"hc4m"`: no method-specific arguments.
- `"hc5"`: `k = 0.7`.
- `"hc5m"`: `k = 0.7`, `k1 = 1`, `k2 = 0`, `k3 = 1`, `gamma1 = 1`, and `gamma2 = 1.5`.
- `"hcbeta"`: `c1 = 7`, `c2 = 0.75`, `lower = 0.01`, `upper = 0.99`, `a_max = 10000`, and `b_max = 10000`.

For `"hc5"` and `"hc5m"`, `k`, `k1`, `k2`, and `k3` must be nonnegative, while `gamma1` and `gamma2` must be positive. For `"hcbeta"`, `c1` must be nonnegative, `c2` must be positive, and `lower` and `upper` must lie in `(0, 1)` with `lower < upper`. The HCbeta leverage-complement truncation is $w_t = max(lower, min(1 - h_t, upper))$.

After shrinkage with $\zeta = n/(n + 50)$, the Beta shape parameters are clamped as

$$\tilde{a} = \min\{\max\{(1-\zeta) + \zeta\hat{a}, \epsilon\}, A_{\max}\},$$

$$\tilde{b} = \min\{\max\{(1-\zeta) + \zeta\hat{b}, \epsilon\}, B_{\max}\}.$$

The value $\epsilon = 0.01$ is fixed, applied after shrinkage and before the caps, and is not a method argument. It is distinct from `lower`, which truncates w_t; changing `lower` does not change ϵ.

Both `a_max` and `b_max` must be finite and lie in `[50, 25000]`; they default to 10000 and can be set independently through ... for sensitivity analysis. When the variance of the truncated complements is numerically degenerate, the adjusted shapes are set directly to their respective caps. HCbeta remains defined when $h_t = 1$ because it truncates $1 - h_t$ before evaluating the Beta CDF. In contrast, HC2 through HC5m require a strictly positive leverage complement.

Value

An object of class `hcinfer_vcov`. The covariance matrix is stored in `object$vcov` and is returned directly by `vcov()`.

References

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4), 817-838. doi:10.2307/1912934

Hinkley, D. V. (1977). Jackknifing in unbalanced situations. *Technometrics*, 19(3), 285-292. doi:10.1080/00401706.1977.10489550

MacKinnon, J. G. and White, H. (1985). Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, 29(3), 305-325. doi:10.1016/03044076(85)901587

Davidson, R. and MacKinnon, J. G. (1993). *Estimation and Inference in Econometrics*. Oxford University Press.

Cribari-Neto, F. (2004). Asymptotic inference under heteroskedasticity of unknown form. *Computational Statistics and Data Analysis*, 45(2), 215-233. doi:10.1016/S01679473(02)003663

Cribari-Neto, F. and da Silva, W. B. (2011). A new heteroskedasticity consistent covariance matrix estimator for the linear regression model. *AStA Advances in Statistical Analysis*, 95(2), 129-146. doi:10.1007/s1018201001412

Cribari-Neto, F., Souza, T. C., and Vasconcellos, K. L. P. (2007). Inference under heteroskedasticity and leveraged data. *Communications in Statistics - Theory and Methods*, 36(10), 1877-1888. doi:10.1080/03610920601126589

Li, S., Zhang, N., Zhang, X., and Wang, G. (2016). A new heteroskedasticity-consistent covariance matrix estimator and inference under heteroskedasticity. *Journal of Statistical Computation and Simulation*, 87(1), 198-210. doi:10.1080/00949655.2016.1198906

Examples

```
schools <- PublicSchools |>
  dplyr::mutate(
    income_scaled = income / 10000,
    income_scaled_sq = income_scaled^2
  )
fit <- lm(expenditure ~ income_scaled + income_scaled_sq, data = schools)
cov <- vcov_hc(fit, type = "hcbeta")
cov
vcov(cov)
plot(cov)

# Sensitivity analysis with nondefault HCbeta caps
```

```
vcov_hc(fit, type = "hcbeta", a_max = 20000, b_max = 20000)
vcov_hc(fit, type = "hc5", k = 0.7)
vcov_hc(fit, type = "hc5m", k = 0.7, k1 = 1, k2 = 0, k3 = 1)
```

Index