Package 'ImpShrinkage'

June 21, 2023

Type Package

Title Improved Shrinkage Estimations for Multiple Linear Regression

Version 1.0.0

Maintainer Mina Norouzirad <mina.norouzirad@gmail.com>

Description
A variety of improved shrinkage estimators in the area of statistical analysis: unrestricted; restricted; preliminary test; improved preliminary test; Stein; and positive-rule Stein. More details can be found in chapter 7 of Saleh, A. K. Md. E. (2006) <ISBN: 978-0-471-56375-4>.

License GPL (>= 2)

URL https://github.com/mnrzrad/ImpShrinkage

Encoding UTF-8

LazyData false

RoxygenNote 7.2.3

Depends R (>= 2.10)

NeedsCompilation no

Author Mina Norouzirad [aut, cre] (<https://orcid.org/0000-0003-0311-6888>),
Danial Mazarei [aut] (<https://orcid.org/0000-0002-3633-9298>),
Ricardo Moura [aut] (<https://orcid.org/0000-0002-3003-9235>),
'FCT, I.P. [fnd] (under the scope of the projects UIDB/00297/2020 and
UIDP/00297/2020 (NovaMath))

Repository CRAN

Date/Publication 2023-06-21 19:50:02 UTC

R topics documented:

`cement` *Hald's Cement Data*

Description

Heat evolved (cals/gm) in the setting of 13 samples of Portland cement with different percentage weight of chemical components.

Format

A data.frame with 13 observations on the following 5 variables.

x1 percentage weight in clinkers of 3CaO.Al2O3

x2 percentage weight in clinkers of 3CaO.SiO2

x3 percentage weight in clinkers of 4CaO.Al2O3.Fe2O3

x4 percentage weight in clinkers of 2CaO.SiO2

y heat evolved (calories/gram)

Source

Woods, H., Steinour, H. H. and Starke, H. R. (1932) Effect of composition of Portland cement on heat evolved during hardening. Industrial Engineering and Chemistry, 24, 1207–1214.

Examples

```
data("cement")
cement
```

`coefficients.improvedpreliminaryTest` *Extract Model Coefficients*

Description

Coefficients extracted from the model object `improvedpreliminaryTest`

Usage

```
## S3 method for class 'improvedpreliminaryTest'
coefficients(object, ...)

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

Arguments

`object`	An object of class `improvedpreliminaryTest`.
`...`	Other arguments.

Value

A vector of coefficients.

See Also

`coefficients.unrestricted`, `coefficients.restricted`, `coefficients.preliminaryTest`, `coefficients.stein`, `coefficients.positivestein`, `coef.unrestricted`, `coef.restricted`, `coef.positivestein`, `coef.stein`, `coef.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- iptReg(X, y, H, h, alpha = 0.05)
coefficients(model)
coef(model)
```

---

coefficients.positivestein

Extract Model Coefficients

Description

Coefficients extracted from the model object `positivestein`

Usage

```
## S3 method for class 'positivestein'
coefficients(object, ...)

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

Arguments

`object`	An object of class `positivestein`.
`...`	Other arguments.

Value

A vector of coefficients.

See Also

`coefficients.unrestricted`, `coefficients.restricted`, `coefficients.preliminaryTest`, `coefficients.improvedpreliminaryTest`, `coefficients.stein`, `coef.unrestricted`, `coef.restricted`, `coef.preliminaryTest`, `coef.improvedpreliminaryTest`, `coef.stein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- prstReg(X, y, H, h)
coefficients(model)
coef(model)
```

---

coefficients.preliminaryTest

Extract Model Coefficients

Description

Coefficients extracted from the model object `preliminaryTest`

Usage

```
## S3 method for class 'preliminaryTest'
coefficients(object, ...)

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

Arguments

`object`	An object of class `preliminaryTest`.
`...`	Other arguments.

Value

A vector of coefficients.

See Also

`coefficients.unrestricted`, `coefficients.restricted`, `coefficients.improvedpreliminaryTest`, `coefficients.stein`, `coefficients.positivestein`, `coef.unrestricted`, `coef.restricted`, `coef.improvedpreliminaryTest`. `coef.stein`, `coef.positivestein`. #'

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- ptReg(X, y, H, h, alpha = 0.05)
coefficients(model)
coef(model)
```

---

coefficients.restricted

Extract Model Coefficients

Description

Coefficients extracted from the model object `restrcited`.

Usage

```
## S3 method for class 'restricted'
coefficients(object, ...)

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

Arguments

`object`	An object of class `restricted`.
`...`	Other arguments.

Value

A vector of coefficients.

See Also

`coefficients.unrestricted`, `coefficients.preliminaryTest`, `coefficients.improvedpreliminaryTest`, `coefficients.stein`, `coefficients.positivestein`, `coef.unrestricted`, `coef.preliminaryTest`, `coef.improvedpreliminaryTest` `coef.stein`, `coef.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- resReg(X, y, H, h)
coefficients(model)
coef(model)
```

`coefficients.stein` *Extract Model Coefficients*

Description

Coefficients extracted from the model object `stein`

Usage

```
## S3 method for class 'stein'
coefficients(object, ...)

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

Arguments

`object`	An object of class `stein`.
`...`	Other arguments.

Value

A vector of coefficients.

See Also

`coefficients.unrestricted`, `coefficients.restricted`, `coefficients.preliminaryTest`, `coefficients.improvedpreliminaryTest`, `coefficients.positivestein`, `coef.unrestricted`, `coef.restricted`, `coef.preliminaryTest`, `coef.improvedpreliminaryTest`, `coef.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nr = 3, nc = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- stReg(X, y, H, h)
coefficients(model)
coef(model)
```

---

coefficients.unrestricted

Extract Model Coefficients

Description

Coefficients extracted from the model object `unrestricted`.

Usage

```
## S3 method for class 'unrestricted'
coefficients(object, ...)

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

Arguments

`object`	An object of class `unrestricted`.
`...`	Other arguments.

Value

A vector of coefficients.

See Also

`coefficients.restricted`, `coefficients.preliminaryTest`, `coefficients.improvedpreliminaryTest`, `coefficients.stein`, `coefficients.positivestein`, `coef.restricted`, `coef.preliminaryTest`, `coef.improvedpreliminaryTest` `coef.stein`, `coef.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
model <- unrReg(X, y)
coefficients(model)
coef(model)
```

`fitted.improvedpreliminaryTest` *Extract Model Fitted Values*

Description

Fitted values based on object `improvedpreliminaryTest`.

Usage

```
## S3 method for class 'improvedpreliminaryTest'
fitted(object, ...)
```

Arguments

`object`	An object of class `improvedpreliminaryTest`.
`...`	Other arguments.

Value

A vector of fitted values.

See Also

`fitted.unrestricted`, `fitted.restricted`, `fitted.preliminaryTest`, `fitted.stein`, `fitted.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
```

```
model <- iptReg(X, y, H, h, alpha = 0.05)
fitted(model)
```

`fitted.positivestein` *Extract Model Fitted Values*

Description

Fitted values based on object `positivestein`.

Usage

```
## S3 method for class 'positivestein'
fitted(object, ...)
```

Arguments

`object`	An object of class `positivestein`.
`...`	Other arguments.

Value

A vector of fitted values.

See Also

`fitted.unrestricted`, `fitted.restricted`, `fitted.preliminaryTest`, `fitted.improvedpreliminaryTest`, `fitted.stein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- prstReg(X, y, H, h)
fitted(model)
```

`fitted.preliminaryTest` *Extract Model Fitted Values*

Description

Fitted values based on object `preliminaryTest`.

Usage

```
## S3 method for class 'preliminaryTest'
fitted(object, ...)
```

Arguments

`object`	An object of class `preliminaryTest`.
`...`	Other arguments.

Value

A vector of fitted values.

See Also

`fitted.unrestricted`, `fitted.restricted`, `fitted.improvedpreliminaryTest`, `fitted.stein`, `fitted.positivestein`

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- ptReg(X, y, H, h, alpha = 0.05)
fitted(model)
```

`fitted.restricted` *Extract Model Fitted Values*

Description

Fitted values based on object `restrcited`.

Usage

```
## S3 method for class 'restricted'
fitted(object, ...)
```

Arguments

`object`	An object of class `restricted`.
`...`	Other arguments.

Value

Fitted values extracted from the object `restricted`.

See Also

`fitted.unrestricted`, `fitted.preliminaryTest`, `fitted.improvedpreliminaryTest`, `fitted.stein`, `fitted.positivestein`

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- resReg(X, y, H, h)
fitted(model)
```

`fitted.stein` *Extract Model Fitted Values*

Description

Fitted values based on object `stein`.

Usage

```
## S3 method for class 'stein'
fitted(object, ...)
```

Arguments

`object`	An object of class `stein`.
`...`	Other arguments.

Value

A vector of fitted values.

See Also

`fitted.unrestricted`, `fitted.restricted`, `fitted.preliminaryTest`, `fitted.improvedpreliminaryTest`, `fitted.positivestein`

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nr = 3, nc = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- stReg(X, y, H, h)
fitted(model)
```

`fitted.unrestricted` *Extract Model Fitted Values*

Description

Fitted values based on object `unrestricted`.

Usage

```
## S3 method for class 'unrestricted'
fitted(object, ...)
```

Arguments

`object`	An object of class `unrestricted`.
`...`	Other arguments.

Value

A vector of fitted values.

See Also

`fitted.restricted`, `fitted.preliminaryTest`, `fitted.improvedpreliminaryTest`, `fitted.stein`, `fitted.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
model <- unrReg(X, y)
fitted(model)
```

iptReg *The improved preliminary test estimator*

Description

This function calculates the improved preliminary test estimator. When the error has a normal distribution, this estimator can be calculated by

$$\hat{\beta}^{iPT} = \hat{\beta}^{PT} - d(\hat{\beta}^{U} - \hat{\beta}^{R})\mathcal{L}^{-1}I(\mathcal{L} > F_{q,n-p}(\alpha))$$

and, when the error has a non-normal distribution, by

$$\hat{\beta}^{iPT} = \hat{\beta}^{PT} - d(\hat{\beta}^{U} - \hat{\beta}^{R})\mathcal{L}^{-1}I(\mathcal{L} > \chi^2_q(\alpha))$$

where $I(A)$ denotes an indicator function and

- $\hat{\beta}^{PT}$ is the preliminary test estimator; See `ptReg`
- $\hat{\beta}^{U}$ is the unrestricted estimator; See `unrReg`.
- $\hat{\beta}^{R}$ is the restricted estimator; See `resReg`.
- $\mathcal{L}$ is the test statistic. See `teststat`;
- $F_{q,n-p}(\alpha)$ is the upper α level critical value of F-distribution with $(q, n-p)$ degrees of freedom, calculated using `qf`;
- $\chi^2_q(\alpha)$ is the upper α level critical value of χ^2-distribution with q degree of freedom, calculated using `qchisq`;
- d is the shrinkage factor;
- α is the significance level.

Usage

```
iptReg(X, y, H, h, alpha, d = NULL, is_error_normal = FALSE)
```

Arguments

`X`	Matrix with input observations, of dimension `n` x `p`; each row is an observation vector.
`y`	Vector with response observations of size `n`.
`H`	A given `q` x `p` matrix.
`h`	A given `q` x `1` vector.
`alpha`	A given significance level.
`d`	(optional) If not provided (or set to `NULL`), it will be calculated using $\frac{(q-2)\cdot(n-p)}{q\cdot(n-p+2)}$.
`is_error_normal`	logical value indicating whether the errors follow a normal distribution. If `is_error_normal` is `TRUE`, the distribution of the test statistics for the null hypothesis is F distribution (`FDist`). On the other hand, if the errors have a non-normal distribution, the asymptotic distribution of the test statistics is χ^2 distribution (`Chisquare`). By default, `is_error_normal` is set to `FALSE`.

Details

The corresponding estimator of σ^2 is

$$s^2 = \frac{1}{n-p}(y - X\hat{\beta}^{iPT})^\top (y - X\hat{\beta}^{iPT}).$$

Value

An object of class `improvedpreliminaryTest` is a list containing at least the following components:

`coef` A named vector of coefficients.

`residuals` The residuals, that is, the response values minus fitted values.

`s2` The estimated variance.

`fitted.values` The fitted values.

References

Saleh, A. K. Md. Ehsanes. (2006). *Theory of Preliminary Test and Stein-Type Estimation With Applications*, Wiley.

Kaciranlar, S., Akdeniz, S. S. F., Styan, G. P. & Werner, H. J. (1999). A new biased estimators in linear regression and detailed analysis of the widely-analysed dataset on portland cement. *Sankhya, Series B*, 61(3), 443-459.

Kibria, B. M. Golam (2005). Applications of Some Improved Estimators in Linear Regression, *Journal of Modern Applied Statistical Methods*, 5(2), 367- 380.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
iptReg(X, y, H, h, alpha = 0.05)

# H beta != h
p <- ncol(X)
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(1, nrow(H))
iptReg(X, y, H, h, alpha = 0.05)

data(cement)
X <- as.matrix(cbind(1, cement[, 1:4]))
y <- cement$y
# Based on Kaciranlar et al. (1999)
```

```
H <- matrix(c(0, 1, -1, 1, 0), nrow = 1, ncol = 5, byrow = TRUE)
h <- rep(0, nrow(H))
iptReg(X, y, H, h, alpha = 0.05)
# Based on Kibria (2005)
H <- matrix(c(0, 1, -1, 1, 0, 0, 0, 1, -1, -1, 0, 1, -1, 0, -1), nrow = 3, ncol = 5, byrow = TRUE)
h <- rep(0, nrow(H))
iptReg(X, y, H, h, alpha = 0.05)
```

`predict.improvedpreliminaryTest` *Extract Model Predictions Values*

Description

Predicted values based on object `improvedpreliminaryTest`.

Usage

```
## S3 method for class 'improvedpreliminaryTest'
predict(object, newdata, ...)
```

Arguments

`object`	An object of class "`improvedpreliminaryTest`".
`newdata`	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
`...`	Other arguments.

Value

A vector of predictions.

See Also

`predict.unrestricted`, `predict.restricted`, `predict.preliminaryTest`, `predict.stein`, `predict.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- iptReg(X, y, H, h, alpha = 0.05)
```

```
predict(model, X)
```

`predict.positivestein` *Extract Model Predictions Values*

Description

Predicted values based on object `positivestein`.

Usage

```
## S3 method for class 'positivestein'
predict(object, newdata, ...)
```

Arguments

`object`	An object of class "`positivestein`".
`newdata`	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
`...`	Other arguments.

Value

A vector of predictions.

See Also

`predict.unrestricted`, `predict.restricted`, `predict.preliminaryTest`, `predict.improvedpreliminaryTest`, `predict.stein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- prstReg(X, y, H, h)
predict(model, X)
```

predict.preliminaryTest *Extract Model Predictions Values*

Description

Predicted values based on object `preliminaryTest`.

Usage

```
## S3 method for class 'preliminaryTest'
predict(object, newdata, ...)
```

Arguments

`object`	An object of class "`preliminaryTest`".
`newdata`	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
`...`	Other arguments.

Value

A vector of predictions.

See Also

`predict.unrestricted`, `predict.restricted`, `predict.improvedpreliminaryTest`, `predict.stein`, `predict.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- ptReg(X, y, H, h, alpha = 0.05)
predict(model, X)
```

`predict.restricted` *Extract Model Predictions Values*

Description

Predicted values based on object `restrcited`.

Usage

```
## S3 method for class 'restricted'
predict(object, newdata, ...)
```

Arguments

`object`	An object of class `restricted`.
`newdata`	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
`...`	Other arguments.

Value

A vector of predictions.

See Also

`predict.unrestricted`, `predict.preliminaryTest`, `predict.improvedpreliminaryTest`, `predict.stein`, `predict.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- resReg(X, y, H, h)
predict(model, X)
```

predict.stein *Extract Model Predictions Values*

Description

Predicted values based on object `stein`.

Usage

```
## S3 method for class 'stein'
predict(object, newdata, ...)
```

Arguments

`object`	An object of class "`stein`".
`newdata`	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
`...`	Other arguments.

Value

A vector of predictions.

See Also

`predict.unrestricted`, `predict.restricted`, `predict.preliminaryTest`, `predict.improvedpreliminaryTest`, `predict.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nr = 3, nc = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- stReg(X, y, H, h)
predict(model, X)
```

`predict.unrestricted` *Extract Model Predictions Values*

Description

Predicted values based on object `unrestricted`.

Usage

```
## S3 method for class 'unrestricted'
predict(object, newdata, ...)
```

Arguments

`object`	An object of class `unrestricted`.
`newdata`	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
`...`	Other arguments.

Value

A vector of predictions.

See Also

`predict.restricted`, `predict.preliminaryTest`, `predict.improvedpreliminaryTest`, `predict.stein`, `predict.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
model <- unrReg(X, y)
predict(model, X)
```

prstReg *The positive-rule Stein estimator*

Description

This function calculates the positive-rule Stein estimator. This estimator is an improved version of the Stein estimator, where only the positive part of the shrinking factor is considered. It may be calculated by

$$\hat{\beta}^{S+} = \hat{\beta}^{S} + (1 + d\mathcal{L}^{-1})I(\mathcal{L} > d)(\hat{\beta}^{U} - \hat{\beta}^{R})$$

where $I(A)$ denotes an indicator function and

- $\hat{\beta}^{S}$ is the Stein estimator; See stReg.
- $\hat{\beta}^{U}$ is the unrestricted estimator; See unrReg.
- $\hat{\beta}^{R}$ is the restricted estimator; See resReg.
- $\mathcal{L}$ is the test statistic. See teststat;
- d is the shrinkage factor.

Usage

```
prstReg(X, y, H, h, d = NULL, is_error_normal = FALSE)
```

Arguments

X	Matrix with input observations, of dimension n x p; each row is an observation vector.
y	Vector with response observations of size n.
H	A given q x p matrix.
h	A given q x 1 vector.
d	(optional) If not provided (or set to NULL), it will be calculated using $\frac{(q-2)\cdot(n-p)}{q\cdot(n-p+2)}$.
is_error_normal	logical value indicating whether the errors follow a normal distribution. If is_error_normal is TRUE, the distribution of the test statistics for the null hypothesis is F distribution (FDist). On the other hand, if the errors have a non-normal distribution, the asymptotic distribution of the test statistics is χ^2 distribution (Chisquare). By default, is_error_normal is set to FALSE.

Details

The corresponding estimator of σ^2 is given by

$$s^2 = \frac{1}{n-p}(y - X\hat{\beta}^{S+})^{\top}(y - X\hat{\beta}^{S+}).$$

Value

An object of class `pst` is a list containing at least the following components:

`coef` A named vector of coefficients.

`residuals` The residuals, that is, the response values minus fitted values.

`s2` The estimated variance.

`fitted.values` The fitted values.

References

Saleh, A. K. Md. Ehsanes. (2006). *Theory of Preliminary Test and Stein-Type Estimation With Applications*, Wiley.

Kaciranlar, S., Akdeniz, S. S. F., Styan, G. P. & Werner, H. J. (1999). A new biased estimators in linear regression and detailed analysis of the widely-analysed dataset on portland cement. *Sankhya, Series B*, 61(3), 443-459.

Kibria, B. M. Golam (2005). Applications of Some Improved Estimators in Linear Regression, *Journal of Modern Applied Statistical Methods*, 5(2), 367- 380.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nr = 3, nc = p, byrow = TRUE)
h <- rep(0, nrow(H))
prstReg(X, y, H, h)

# H beta != h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nr = 3, nc = p, byrow = TRUE)
h <- rep(1, nrow(H))
prstReg(X, y, H, h)

data(cement)
X <- as.matrix(cbind(1, cement[, 1:4]))
y <- cement$y
# Based on Kaciranlar et al. (1999)
H <- matrix(c(0, 1, -1, 1, 0), nrow = 1, ncol = 5, byrow = TRUE)
h <- rep(0, nrow(H))
prstReg(X, y, H, h)
# Based on Kibria (2005)
H <- matrix(c(0, 1, -1, 1, 0, 0, 0, 1, -1, -1, 0, 1, -1, 0, -1), nrow = 3, ncol = 5, byrow = TRUE)
h <- rep(0, nrow(H))
prstReg(X, y, H, h)
```

ptReg *The Preliminary Test Estimator*

Description

This function calculates the preliminary test. When the error has a normal distribution, the test statistic is given by

$$\hat{\beta}^{PT} = \hat{\beta}^{U} - (\hat{\beta}^{U} - \hat{\beta}^{R})I(\mathcal{L} \leq F_{q,n-p}(\alpha))$$

and, if the error has a non-normal distribution, is given by

$$\hat{\beta}^{PT} = \hat{\beta}^{U} - (\hat{\beta}^{U} - \hat{\beta}^{R})I(\mathcal{L} \leq \chi^2_q(\alpha))$$

where $I(A)$ denotes an indicator function and

- $\hat{\beta}^{U}$ is the unrestricted estimator; See unrReg.
- $\hat{\beta}^{R}$ is the restricted estimator; See resReg.
- $\mathcal{L}$ is the test statistic. See teststat;
- $F_{q,n-p}(\alpha)$ is the upper α level critical value of F-distribution with $(q, n-p)$ degrees of freedom, calculated using qf;
- $\chi^2_q(\alpha)$is the upper α level critical value of χ^2-distribution with q degree of freedom, calculated using qchisq;
- α: the significance level.

Usage

```
ptReg(X, y, H, h, alpha, is_error_normal = FALSE)
```

Arguments

X	Matrix with input observations, of dimension n x p; each row is an observation vector.
y	Vector with response observations of size n.
H	A given q x p matrix.
h	A given q x 1 vector.
alpha	A given significance level.
is_error_normal	logical value indicating whether the errors follow a normal distribution. If is_error_normal is TRUE, the distribution of the test statistics for the null hypothesis is F distribution (FDist). On the other hand, if the errors have a non-normal distribution, the asymptotic distribution of the test statistics is χ^2 distribution (Chisquare). By default, is_error_normal is set to FALSE.

Details

The corresponding estimator of σ^2 is

$$s^2 = \frac{1}{n-p}(y - X\hat{\beta}^{PT})^\top (y - X\hat{\beta}^{PT}).$$

Value

An object of class `preliminaryTest` is a list containing at least the following components:

`coef` A named vector of coefficients.

`residuals` The residuals, that is, the response values minus fitted values.

`s2` The estimated variance.

`fitted.values` The fitted values.

References

Saleh, A. K. Md. Ehsanes. (2006). *Theory of Preliminary Test and Stein-Type Estimation With Applications*, Wiley.

Kaciranlar, S., Akdeniz, S. S. F., Styan, G. P. & Werner, H. J. (1999). A new biased estimators in linear regression and detailed analysis of the widely-analysed dataset on portland cement. *Sankhya, Series B*, 61(3), 443-459.

Kibria, B. M. Golam (2005). Applications of Some Improved Estimators in Linear Regression, *Journal of Modern Applied Statistical Methods*, 5(2), 367- 380.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
ptReg(X, y, H, h, alpha = 0.05)

# H beta != h
p <- ncol(X)
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(1, nrow(H))
ptReg(X, y, H, h, alpha = 0.05)

data(cement)
X <- as.matrix(cbind(1, cement[, 1:4]))
y <- cement$y
# Based on Kaciranlar et al. (1999)
H <- matrix(c(0, 1, -1, 1, 0), nrow = 1, ncol = 5, byrow = TRUE)
```

```
h <- rep(0, nrow(H))
ptReg(X, y, H, h, alpha = 0.05)
# Based on Kibria (2005)
H <- matrix(c(0, 1, -1, 1, 0, 0, 0, 1, -1, -1, 0, 1, -1, 0, -1), nrow = 3, ncol = 5, byrow = TRUE)
h <- rep(0, nrow(H))
ptReg(X, y, H, h, alpha = 0.05)
```

`residuals.improvedpreliminaryTest` *Extract Model Residuals*

Description

Residuals values based on model object `improvedpreliminaryTest`.

Usage

```
## S3 method for class 'improvedpreliminaryTest'
residuals(object, ...)
```

Arguments

`object`	An object of class `improvedpreliminaryTest`.
`...`	Other arguments.

Value

A vector of residuals.

See Also

`residuals.unrestricted`, `residuals.restricted`, `residuals.preliminaryTest`, `residuals.stein`, `residuals.positivestein`,

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- iptReg(X, y, H, h, alpha = 0.05)
residuals(model)
```

`residuals.positivestein` *Extract Model Residuals*

Description

Residuals values based on model object `positivestein`.

Usage

```
## S3 method for class 'positivestein'
residuals(object, ...)
```

Arguments

`object`	An object of class `positivestein`.
`...`	Other arguments.

Value

A vector of residuals.

See Also

`residuals.unrestricted`, `residuals.restricted`, `residuals.preliminaryTest`, `residuals.improvedpreliminary`
`residuals.stein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- prstReg(X, y, H, h)
residuals(model)
```

`residuals.preliminaryTest` *Extract Model Residuals*

Description

Residuals values based on model object `preliminaryTest`.

Usage

```
## S3 method for class 'preliminaryTest'
residuals(object, ...)
```

Arguments

`object`	An object of class `preliminaryTest`.
`...`	Other arguments.

Value

A vector of residuals.

See Also

`residuals.unrestricted`, `residuals.restricted`, `residuals.improvedpreliminaryTest`, `residuals.stein`, `residuals.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- ptReg(X, y, H, h, alpha = 0.05)
residuals(model)
```

residuals.restricted *Extract Model Residuals*

Description

Residuals values based on model object `restricted`.

Usage

```
## S3 method for class 'restricted'
residuals(object, ...)
```

Arguments

`object`	An object of class `restricted`.
`...`	Other arguments.

Value

A vector of residuals.

`residuals.unrestricted`, `residuals.preliminaryTest`, `residuals.improvedpreliminaryTest`, `residuals.stein`, `residuals.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- resReg(X, y, H, h)
residuals(model)
```

residuals.stein *Extract Model Residuals*

Description

Residuals values based on model object `stein`.

Usage

```
## S3 method for class 'stein'
residuals(object, ...)
```

Arguments

object	An object of class stein.
...	Other arguments.

Value

A vector of residuals.

See Also

residuals.unrestricted, residuals.restricted, residuals.preliminaryTest, residuals.improvedpreliminary
residuals.positivestein,.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nr = 3, nc = p, byrow = TRUE)
h <- rep(0, nrow(H))
model <- stReg(X, y, H, h)
residuals(model)
```

---

residuals.unrestricted

Extract Model Residuals

Description

Residuals values based on model object unrestricted.

Usage

```
## S3 method for class 'unrestricted'
residuals(object, ...)
```

Arguments

`object`	An object of class `unrestricted`.
`...`	Other arguments.

Value

A vector of residuals.

See Also

`residuals.restricted`, `residuals.preliminaryTest`, `residuals.improvedpreliminaryTest` `residuals.stein`, `residuals.positivestein`.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
model <- unrReg(X, y)
residuals(model)
```

---

resReg *The restricted estimator*

Description

This function calculates the restricted estimator using

$$\hat{\beta}^R = \hat{\beta}^U - (X^\top X)^{-1} H^\top (H(X^\top X)^{-1} H^\top)^{-1} (H\hat{\beta}^U - h)$$

where

- $\hat{\beta}^U$ is the unrestricted estimator; See `unrReg`.
- $H\beta = h$ represents a subspace of the parameter space induced by the non-sample information. Here, H is a known $q \times p$ matrix, and h is a known q-vector.

Usage

```
resReg(X, y, H, h)
```

Arguments

`X`	Matrix with input observations, of dimension `n` x `p`; each row is an observation vector.
`y`	Vector with response observations of size `n`.
`H`	A given `q` x `p` matrix.
`h`	A given `q` x `1` vector.

Details

#' The corresponding estimator of σ^2 is

$$s^2 = \frac{1}{n-p}(y - X\hat{\beta}^R)^\top (y - X\hat{\beta}^R).$$

Value

An object of class `restricted` is a list containing at least the following components:

`coef` A named vector of coefficients.

`residuals` The residuals, that is, the response values minus fitted values.

`s2` The estimated variance.

`fitted.values` The fitted values.

References

Saleh, A. K. Md. Ehsanes. (2006). *Theory of Preliminary Test and Stein-Type Estimation With Applications*, Wiley.

Kaciranlar, S., Akdeniz, S. S. F., Styan, G. P. & Werner, H. J. (1999). A new biased estimators in linear regression and detailed analysis of the widely-analysed dataset on portland cement. *Sankhya, Series B*, 61(3), 443-459.

Kibria, B. M. Golam (2005). Applications of Some Improved Estimators in Linear Regression, *Journal of Modern Applied Statistical Methods*, 5(2), 367- 380.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
resReg(X, y, H, h)

# H beta != h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(1, nrow(H))
resReg(X, y, H, h)


data(cement)
X <- as.matrix(cbind(1, cement[, 1:4]))
y <- cement$y
# Based on Kaciranlar et al. (1999)
H <- matrix(c(0, 1, -1, 1, 0), nrow = 1, ncol = 5, byrow = TRUE)
```

```
h <- rep(0, nrow(H))
resReg(X, y, H, h)
# Based on Kibria (2005)
H <- matrix(c(0, 1, -1, 1, 0, 0, 0, 1, -1, -1, 0, 1, -1, 0, -1), nrow = 3, ncol = 5, byrow = TRUE)
h <- rep(0, nrow(H))
resReg(X, y, H, h)
```

`simdata` *Simulation data*

Description

This function generates a toy example. The error term, ε, and the design matrix, X, are simulated from standard normal distributions, $\mathcal{N}(0, 1)$, using the `rnorm` function. Given the true parameter vector, β, the response vector, y, is calculated as

$$y = X\beta + \varepsilon.$$

Usage

```
simdata(n, p, beta, seed = NULL)
```

Arguments

`n`	Number of observations.
`p`	Number of variables.
`beta`	Regression parameter.
`seed`	(Optional) The random seed for reproducibility. Default is `NULL`.

Value

A list containing the following components:

X a matrix of dimensions `n` x `p`.

y a numeric vector of length `n`.

References

Saleh, A. K. Md. Ehsanes. (2006). *Theory of Preliminary Test and Stein-Type Estimation With Applications*, Wiley.

Examples

```
simulated_data <- simdata(n = 100, p = 5, beta = c(2, 1, 3, 0, 5))
X <- simulated_data$X
y <- simulated_data$y
X
y
```

`stReg` *The Stein estimator*

Description

This function can be used to calculate the Stein estimator using

$$\hat{\beta}^S = \hat{\beta}^U - d\mathcal{L}^{-1}(\hat{\beta}^U - \hat{\beta}^R)$$

where

- $\hat{\beta}^U$ is the unrestricted estimator; See `unrReg`.
- $\hat{\beta}^R$ is the restricted estimator; See `resReg`.
- $\mathcal{L}$ is the test statistic. See `teststat`;
- d is the shrinkage factor.

Usage

```
stReg(X, y, H, h, d = NULL, is_error_normal = FALSE)
```

Arguments

`X`	Matrix with input observations, of dimension `n` x `p`; each row is an observation vector.
`y`	Vector with response observations of size `n`.
`H`	A given `q` x `p` matrix.
`h`	A given `q` x `1` vector.
`d`	(Optional) If not provided (or set to `NULL`), it will be set to be equal to $\frac{(q-2)\cdot(n-p)}{q\cdot(n-p+2)}$.
`is_error_normal`	logical value indicating whether the errors follow a normal distribution. If `is_error_normal` is `TRUE`, the distribution of the test statistics for the null hypothesis is `FDist`. On the other hand, if the errors have a non-normal distribution, the asymptotic distribution of the test statistics is `Chisquare`. By default, `is_error_normal` is set to `FALSE`.

Details

The corresponding estimator of σ^2 is

$$s^2 = \frac{1}{n-p}(y - X\hat{\beta}^S)^\top(y - X\hat{\beta}^S).$$

Value

An object of class `stein` is a list containing at least the following components:

`coef` A vector of coefficients.

`residuals` The residuals, that is, the response values minus the fitted values.

`s2` The estimated variance.

`fitted.values` The fitted values.

References

Saleh, A. K. Md. Ehsanes. (2006). *Theory of Preliminary Test and Stein-Type Estimation With Applications*, Wiley.

Kaciranlar, S., Akdeniz, S. S. F., Styan, G. P. & Werner, H. J. (1999). A new biased estimators in linear regression and detailed analysis of the widely-analysed dataset on portland cement. *Sankhya, Series B*, 61(3), 443-459.

Kibria, B. M. Golam (2005). Applications of Some Improved Estimators in Linear Regression, *Journal of Modern Applied Statistical Methods*, 5(2), 367- 380.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
stReg(X, y, H, h)

# H beta != h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(1, nrow(H))
stReg(X, y, H, h)

data(cement)
X <- as.matrix(cbind(1, cement[, 1:4]))
y <- cement$y
# Based on Kaciranlar et al. (1999)
H <- matrix(c(0, 1, -1, 1, 0), nrow = 1, ncol = 5, byrow = TRUE)
h <- rep(0, nrow(H))
stReg(X, y, H, h)
# Based on Kibria (2005)
H <- matrix(c(0, 1, -1, 1, 0, 0, 0, 1, -1, -1, 0, 1, -1, 0, -1), nrow = 3, ncol = 5, byrow = TRUE)
h <- rep(0, nrow(H))
stReg(X, y, H, h)
```

`teststat` *Test-Statistics*

Description

This function calculates the test statistics, assuming $\mathcal{H}_0 : H\beta = h$. When the error has a normal distribution, it is defined as

$$\mathcal{L} = \frac{(H\hat{\beta}^U - h)^\top (H(X^\top X)^{-1}H^\top)^{-1}(H\hat{\beta}^U - h)}{q s^2_{unr}}$$

and when the error has a non-normal distribution, as

$$\mathcal{L} = \frac{(H\hat{\beta}^U - h)^\top (H(X^\top X)^{-1}H^\top)^{-1}(H\hat{\beta}^U - h)}{s^2_{unr}}$$

where

- $\hat{\beta}^U$ is the unrestricted estimator; See `unrReg`.
- q is the number of restrictions, i.e., the number of rows of known matrix H;
- s^2_{unr} is the corresponding unrestricted estimator of σ^2.

Usage

```
teststat(X, y, H, h, is_error_normal = FALSE)
```

Arguments

`X`	Matrix with input observations, of dimension `n` x `p`; each row is an observation vector.
`y`	Vector with response observations of size `n`.
`H`	A given `q` x `p` matrix.
`h`	A given `q` x `1` vector.
`is_error_normal`	logical value indicating whether the errors follow a normal distribution. If `is_error_normal` is `TRUE`, the distribution of the test statistics for the null hypothesis is the F distribution (`FDist`).On the other hand, if the errors have a non-normal distribution, the asymptotic distribution of the test statistics is the χ^2 distribution (`Chisquare`). By default, `is_error_normal` is set to `FALSE`.

Value

The value of the test statistic.

References

Saleh, A. K. Md. Ehsanes. (2006). *Theory of Preliminary Test and Stein-Type Estimation With Applications*, Wiley.

Kaciranlar, S., Akdeniz, S. S. F., Styan, G. P. & Werner, H. J. (1999). A new biased estimators in linear regression and detailed analysis of the widely-analysed dataset on portland cement. *Sankhya, Series B*, 61(3), 443-459.

Kibria, B. M. Golam (2005). Applications of Some Improved Estimators in Linear Regression, *Journal of Modern Applied Statistical Methods*, 5(2), 367- 380.

Examples

```
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
p <- ncol(X)
# H beta = h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(0, nrow(H))
teststat(X, y, H, h)

# H beta != h
H <- matrix(c(1, 1, -1, 0, 0, 1, 0, 1, 0, -1, 0, 0, 0, 1, 0), nrow = 3, ncol = p, byrow = TRUE)
h <- rep(1, nrow(H))
teststat(X, y, H, h)

data(cement)
X <- as.matrix(cbind(1, cement[, 1:4]))
y <- cement$y
# Based on Kaciranlar et al. (1999)
H <- matrix(c(0, 1, -1, 1, 0), nrow = 1, ncol = 5, byrow = TRUE)
h <- rep(0, nrow(H))
teststat(X, y, H, h)
# Based on Kibria (2005)
H <- matrix(c(0, 1, -1, 1, 0, 0, 0, 1, -1, -1, 0, 1, -1, 0, -1), nrow = 3, ncol = 5, byrow = TRUE)
h <- rep(0, nrow(H))
teststat(X, y, H, h)
```

---

unrReg *The Unrestricted estimator*

Description

This function calculates the unrestricted estimator as

$$\hat{\beta}^U = (X^\top X)^{-1} X^\top y$$

where $^\top$ denotes the transpose of a matrix. It is important to note that the input matrices X and y should be standardized, for example, by using `scale`. Alternatively, the user can employ `lm` to obtain this estimator, but it is crucial to remember to set `intercept = FALSE`.

Usage

```
unrReg(X, y)
```

Arguments

`X`	Matrix with input observations, of dimension `n` x `p`, where each row is an observation vector;
`y`	Vector with response observations of size `n`.

Details

The corresponding unrestricted estimator of σ^2 is

$$s^2 = \frac{1}{n-p}(y - X\hat{\beta}^U)^\top (y - X\hat{\beta}^U).$$

Value

An object of class `unrestricted` is a list containing at least the following components:

`coef` A named vector of coefficients.

`residuals` The residuals, that is, the response values minus fitted values.

`s2` The estimated variance.

`fitted.values` The fitted values.

References

Saleh, A. K. Md. Ehsanes. (2006). *Theory of Preliminary Test and Stein-Type Estimation With Applications*, Wiley.

Examples

```
data(cement)
n_obs <- 100
p_vars <- 5
beta <- c(2, 1, 3, 0, 5)
simulated_data <- simdata(n = n_obs, p = p_vars, beta)
X <- simulated_data$X
y <- simulated_data$y
unrReg(X, y)

data(cement)
X <- as.matrix(cbind(1, cement[, 1:4]))
y <- cement$y
# Based on Kaciranlar et al. (1999)
H <- matrix(c(0, 1, -1, 1, 0), nrow = 1, ncol = 5, byrow = TRUE)
```

```
h <- rep(0, nrow(H))
unrReg(X, y)

H <- matrix(c(0, 1, -1, 1, 0, 0, 0, 1, -1, -1, 0, 1, -1, 0, -1), nrow = 3, ncol = 5, byrow = TRUE)
h <- rep(0, nrow(H))
unrReg(X, y)
```

Index