Package ‘Rfit’

September 8, 2023

Type Package
Title Rank-Based Estimation for Linear Models
Version 0.24.6
Date 2023-09-02
Author John Kloke, Joseph McKeen
Maintainer John Kloke <johndkloke@gmail.com>
Description Rank-based (R) estimation and inference for linear models. Estimation is for general scores and a library of commonly used score functions is included.
License GPL (>= 2)
LazyLoad yes
LazyData yes
Depends methods
Suggests testthat
NeedsCompilation yes
Repository CRAN
Date/Publication 2023-09-08 05:10:02 UTC

R topics documented:

Rfit-package ......................................................... 2
allscores ............................................................ 3
baseball ............................................................. 4
bbsalaries ........................................................... 5
BoxCox ............................................................... 6
CardioRiskFactors .................................................. 7
confintadjust ........................................................ 8
disp ................................................................. 9
drop.test ........................................................... 10
ffa ................................................................. 11
getScores-methods .................................................. 12
Description

Package provides functions for rank-based analyses of linear models. Rank-based estimation and inference offers a robust alternative to least squares.

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>Rfit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>0.24.2</td>
</tr>
<tr>
<td>Date</td>
<td>2019-12-18</td>
</tr>
<tr>
<td>License</td>
<td>GPL (version 2 or later)</td>
</tr>
<tr>
<td>LazyLoad</td>
<td>yes</td>
</tr>
</tbody>
</table>
Author(s)

John Kloke, Joseph McKean
Maintainer: John Kloke <johndkloke@gmail.com>

References


Examples

data(baseball)
data(wscores)
fit<-rfit(weight~height,data=baseball)
summary(fit)
plot(fitted(fit),rstudent(fit))

### Example of the Reduction (Drop) in dispersion test ###
y<-rnorm(47)
x1<-rnorm(47)
x2<-rnorm(47)
fitF<-rfit(y~x1+x2)
fitR<-rfit(y~x1)
drop.test(fitF,fitR)

---

allscores All Scores

Description

An object of class scores which includes the score function and it’s derivative for rank-based regression inference.

Usage

data(wscores)

Format

The format is: Formal class ‘scores’ [package ".GlobalEnv"] with 2 slots ..@ phi: function (u) ..@ Dphi: function (u)
Details

Using Wilcoxon (linear) scores leads to inference which has ARE of 0.955 to least squares (ML) when the data are normal. Wilcoxon scores are optimal when the underlying error distribution is logistic. Normal scores are optimal when the data are normally distributed. Log-rank scores are optimal when the data are from an exponential distribution, e.g. in a proportional hazards model. Log-Generalized F scores can also be used in the analysis of survival data (see Hettmansperger and McKean p. 233).

bentscores1 are recommended for right-skewed distributions. bentscores2 are recommended for light-tailed distributions. bentscores3 are recommended for left-skewed distributions. bentscores4 are recommended for heavy-tailed distributions.

References


Examples

```r
u <- seq(0.01,0.99,by=0.01)
plot(u,getScores(wscores,u),type='l',main='Wilcoxon Scores')
plot(u,getScores(nscores,u),type='l',main='Normal Scores')
data(wscores)
x<-runif(50)
y<-rlogis(50)
rfit(y~x,scores=wscores)
x<-rnorm(50)
y<-rnorm(50)
rfit(y~x,scores=nscores)
```

---

baseball  
*Baseball Card Data*

Description

These data come from the back-side of 59 baseball cards that Carrie had.

Usage

```r
data(baseball)
```
**bbsalaries**

**Format**

A data frame with 59 observations on the following 6 variables.

- **height**  Height in inches
- **weight**  Weight in pounds
- **bat**  a factor with levels L R S
- **throw**  a factor with levels L R
- **field**  a factor with levels 0 1
- **average**  ERA if the player is a pitcher and his batting average if the player is a fielder

**Source**


**Examples**

```r
data(baseball)
wilcox.test(height~field,data=baseball)
rfit(weight~height,data=baseball)
```

---

**bbsalaries** *Baseball Salaries*

**Description**

Salaries of 176 professional baseball players for the 1987 season.

**Usage**

```r
data(bbsalaries)
```

**Format**

A data frame with 176 observations on the following 8 variables.

- **logYears**  Log of the number of years experience
- **aveWins**  Average wins per year
- **aveLosses**  Average losses per year
- **era**  Earned Run Average
- **aveGames**  Average games pitched in per year
- **aveInnings**  Average number of innings pitched per year
- **aveSaves**  Average number of saves per year
- **logSalary**  Log of the base salary in dollars
BoxCox

Source

http://lib.stat.cmu.edu/datasets/baseball.data

References


Examples

data(bbsalaries)
summary(rfit(logSalary~logYears+aveWins+aveLosses+era+aveGames+aveInnings+aveSaves,data=bbsalaries))

BoxCox

Box and Cox (1964) data.

Description

The data are the results of a 3 * 4 two-way design, where forty-eight animals were exposed to three different poisons and four different treatments. The design is balanced with four replications per cell. The response was the log survival time of the animal.

Usage

data(BoxCox)

Format

A data frame with 48 observations on the following 3 variables.

<table>
<thead>
<tr>
<th>logSurv</th>
<th>log Survival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poison</td>
<td>a factor indicating poison level</td>
</tr>
<tr>
<td>Treatment</td>
<td>a factor indicating treatment level</td>
</tr>
</tbody>
</table>

Source


References


Examples

data(BoxCox)
with(BoxCox,interaction.plot(Treatment,Poison,logSurv,median))
raov(logSurv~Poison+Treatment,data=BoxCox)
**CardioRiskFactors**  

**Cardiovascular risk factors**

**Description**

Data from a study to investigate association between uric acid and various cardiovascular risk factors in developing countries (Heritier et. al. 2009). There are 474 men and 524 women aged 25-64.

**Usage**

```r
data(CardioRiskFactors)
```

**Format**

A data frame with 998 observations on the following 14 variables.

- **age**  Age of subject
- **bmi**  Body Mass Index
- **waisthip**  waist/hip ratio
- **smok**  indicator for regular smoker
- **choles**  total cholesterol
- **trig**  triglycerides level in body fat
- **hdl**  high-density lipoprotein
- **ldl**  low-density lipoprotein
- **sys**  systolic blood pressure
- **dia**  diastolic blood pressure
- **Uric**  serum uric
- **sex**  indicator for male
- **alco**  alcohol intake (mL/day)
- **apoa**  apoprotein A

**Details**


**Source**


Examples

data(CardioRiskFactors)
fitF<-rfit(Uric~bmi+sys+choles+ldl+sex+smok+alco+apoa+trig+age,data=CardioRiskFactors)
fitR<-rfit(Uric~bmi+sys+choles+ldl+sex,data=CardioRiskFactors)
drop.test(fitF,fitR)
summary(fitR)

confintadjust

**Confidence interval adjustment methods**

Description

Returns the critical value to be used in calculating adjusted confidence intervals. Currently provides methods for Bonferroni and Tukey for confidence interval adjustment methods as well as no adjustment.

Usage

confintadjust(n, k, alpha = 0.05, method = confintadjust.methods, ...)

Arguments

- **n**  sample size
- **k**  number of comparisons
- **alpha**  overall (experimentwise) type I error rate
- **method**  one of confintadjust.methods
- **...**  Additional arguments. Currently not used.

Details

Returns critical value based on one of the adjustment methods.

Value

- **cv**  critical value
- **method**  the method used

Author(s)

Joseph McKean, John Kloke

References

Jaeckel’s Dispersion Function

Description

Returns the value of Jaeckel’s dispersion function for given values of the regression coefficients.

Usage

```
disp(beta, x, y, scores)
```

Arguments

- `beta`: p by 1 vector of regression coefficients
- `x`: n by p design matrix
- `y`: n by 1 response vector
- `scores`: an object of class scores

Details

Returns the value of Jaeckel’s dispersion function evaluated at the value of the parameters in the function call. That is, \( \sum_{i=1}^{n} a(R(e_i)) \times e_i \) where R denotes rank and \( a(1) \leq a(2) \leq \ldots \leq a(n) \) are the scores. The residuals \( (e_i i=1,...,n) \) are calculated \( y - x \beta \).

Author(s)

John Kloke, Joseph McKean

References


See Also

- `rfit`
- `drop.test`
- `summary.rfit`
drop.test  
*Drop (Reduction) in Dispersion Test*

**Description**

Given two full model fits, this function performs a reduction in dispersion test.

**Usage**

```r
drop.test(fitF, fitR = NULL)
```

**Arguments**

- `fitF`  
  An object of class rfit. The full model fit.

- `fitR`  
  An object of class rfit. The reduced model fit.

**Details**

Rank-based inference procedure analogous to the traditional (LS) reduced model test.

The full and reduced model dispersions are calculated. The reduction in dispersion test, or drop test for short, has an asymptotic chi-sq distribution. Simulation studies suggest using F critical values. The p-value returned is based on a F-distribution with df1 and df2 degrees of freedom where df1 is the difference in the number of parameters in the fits of fitF and fitR and df2 is the residual degrees of freedom in the fit fitF.

Both fits are based on a minimization routine. It is possible that resulting solutions are such that the fitF$disp > fitRdisp. We recommend starting the full model at the reduced model fit as a way to avoid this situation. See examples.

Checks to see if models appear to be proper subsets. The space spanned by the columns of the reduced model design matrix should be a subset of the space spanned by the columns of the full model design matrix.

**Value**

- `F`  
  Value of the F test statistic

- `p.value`  
  The observed significance level of the test (using an F quantile)

- `RD`  
  Reduced model dispersion minus Full model dispersion

- `tauhat`  
  Estimate of the scale parameter (using the full model residuals)

- `df1`  
  numerator degrees of freedom

- `df2`  
  denominator degrees of freedom

**Author(s)**

John Kloke, Joseph McKeown
References


See Also

rfit

Examples

```r
y<-rnorm(47)
x1<-rnorm(47)
x2<-rnorm(47)
fitF<-rfit(y~x1+x2)
fitR<-rfit(y~x1)
drop.test(fitF,fitR)

## try starting the full model at the reduced model fit ##
fitF<-rfit(y~x1+x2,yhat0=fitR$fitted)
drop.test(fitF,fitR)
```

ffa  

*Free Fatty Acid Data*

Description

The response variable is level of free fatty acid in a sample of prepubescent boys. The explanatory variables are age (in months), weight (in lbs), and skin fold thickness.

Usage

```r
data(ffa)
```

Format

A data frame with 41 rows and 4 columns.

- **age**  age in years
- **weight**  weight in lbs
- **skin**  skin fold thickness
- **ffa**  free fatty acid

Source

References

Examples

data(ffa)
summary(rfit(ffa~age+weight+skin,data=ffa)) #using the default (Wilcoxon scores)
summary(rfit(ffa~age+weight+weight+skin,data=ffa,scores=bentscores1))

getScores-methods

~~ Methods for Function getScores ~~

Description
~~ Methods for function getScores ~~ Calculates the centered and scaled scores as used in rank-based analysis.

Methods
signature(object = "scores")

See Also
rfit

gscoresDeriv-methods

~~ Methods for Function getscoresDeriv ~~

Description
~~ Methods for function getscoresDeriv ~~ This derivative is used in the estimate of the scale parameter tau.

Methods
signature(object = "scores")

See Also
rfit
Description
An estimate of the scale parameter tau is needed for the standard errors of the coefficients in rank-based regression.

Usage
gettau(ehat, p, scores = Rfit::wscores, delta = 0.8, hparm = 2, ...)

Arguments
- **ehat**: full model residuals
- **p**: number of regression coefficients
- **scores**: object of class scores, defaults to Wilcoxon scores
- **delta**: confidence level
- **hparm**: Joe’s hparm
- **...**: additional arguments, currently unused

Details
This is the confidence interval type estimate of the scale parameter tau developed by Koul, Sievers, and McKean (1987). This estimate is also discussed in Section 3.7.1 of Hettmansperger and McKean (1998). One of these function is called in rfit. The default is to use the faster FORTRAN version. The R version can be more precise in small samples, but also can be much slower especially when sample sizes are large.

Value
Length one numeric object.

Author(s)
Joseph McKean, John Kloke

References

See Also

`rfit`

---

**grad**

*Calculate the Gradient of Jaeckel’s Dispersion Function*

**Description**

Calculate the Gradient of Jaeckel’s Dispersion Function

**Usage**

```
grad(x, y, beta, scores)
```

**Arguments**

- `x`: n by p design matrix
- `y`: n by 1 response vector
- `beta`: p by 1 vector of regression coefficients
- `scores`: an object of class scores

**Value**

The gradient evaluated at beta.

**Author(s)**

John Kloke

**References**


**See Also**

`disp`
Examples

```r
## The function is currently defined as
function (x, y, beta, scores)
{
  x <- as.matrix(x)
  e <- y - x %*% beta
  r <- rank(e, ties.method = "first")(length(e) + 1)
  -t(x) %*% scores@phi(r)
}
```

**jaeckel**  
*Function to Minimize Jaeckel’s Dispersion Function*

Description

Uses the built-in function `optim` to minimize Jaeckel’s dispersion function.

Usage

`jaeckel(x, y, beta0 = lm(y ~ x)$coef[2:(ncol(x) + 1)],
scores = Rfit::wscores, control = NULL,...)`

Arguments

- `x`: n by p design matrix
- `y`: n by 1 response vector
- `beta0`: initial estimate
- `scores`: object of class `scores`
- `control`: control passed to fitting routine
- `...`: additional arguments to be passed to fitting routine

Details

Function uses `optim` with method set to BFGS to minimize Jaeckel’s dispersion function. If control is not specified at the function call, the relative tolerance (reltol) is set to `.Machine$double.eps^0(3/4)` maximum number of iterations is set to 200. See `optim`.

Value

Results of `optim` are returned.

Author(s)

John Kloke
References


See Also

optim, rfit

Examples

```r
## This is a internal function. See rfit for user-level examples.
```

**Description**

These are internal functions used to construct the robust anova table. The function raov is the main program.

**Usage**

```r
kwayr(levs, data)
cellx(X)
khmat(levsind, permh)
pasteColsRfit(x, sep ="")
redmod(xmat, amat)
subsets(k)
```

**Arguments**

- `levs` vector of levels corresponding to each of the factors
- `data` data matrix in the form y, factor 1,..., factor k
- `X` n x k matrix where the columns represent the levels of the k factors.
- `levsind` Internal parameter.
- `permh` Internal parameter.
- `x` n x k matrix where the columns represent the levels of the k factors.
- `xmat` n x p full model design matrix
- `amat` Internal parameter.
- `k` Internal parameter.
- `sep` Separator used in pasteColsRfit
Note

Renamed pasteCols of library plotrix written by Jim Lemon et. al. June 2011 under GPL 2

Author(s)

Joseph McKean, John Kloke

References


See Also

raov

Description

Carries out a robust analysis of variance for a one factor design. Analysis is based on the R estimates.

Usage

```r
oneway.rfit(y, g, scores = Rfit::wscores, p.adjust = "none")
```

Arguments

- `y`: n by 1 response vector
- `g`: n by 1 vector representing group membership
- `scores`: an object of class 'scores'
- `p.adjust`: adjustment to the p-values, argument passed to p.adjust

Details

Carries out a robust one-way analysis of variance based on full model r fit.
**Value**

- **fit**: full model fit from `rfit`
- **est**: Estimates
- **se**: Standard Errors
- **I**: First Index
- **J**: Second Index
- **p.value**: p-values
- **y**: response vector
- **g**: vector denoting group membership

**Author(s)**

Joseph McKean, John Kloke

**References**


**See Also**

`rfit`

**Examples**

```r
data(quail)
oneway.rfit(quail$ldl, quail$treat)
```

---

**Description**

Internal class for use with score functions.

**Objects from the Class**

A virtual Class: No objects may be created from it.

**Methods**

No methods defined with class "param" in the signature.
Author(s)
John Kloke

See Also
scores

Examples
showClass("param")

print.rfit Rfit Internal Print Functions

Description
These functions print the output in a user-friendly manner using the internal R function print.

Usage
## S3 method for class 'rfit'
print(x, ...)
## S3 method for class 'summary.rfit'
print(x, digits = max(5, .Options$digits - 2), ...)
## S3 method for class 'drop.test'
print(x, digits = max(5, .Options$digits - 2), ...)
## S3 method for class 'oneway.rfit'
print(x, digits = max(5, .Options$digits - 2), ...)
## S3 method for class 'summary.oneway.rfit'
print(x, digits = max(5, .Options$digits - 2), ...)
## S3 method for class 'raov'
print(x, digits = max(5, .Options$digits - 2), ...)

Arguments
x An object to be printed
digits number of digits to display
... additional arguments to be passed to print

Author(s)
John Kloke

See Also
rfit, summary.rfit, drop.test
**quail**  
*Quail Data*

**Description**  
Thirty-nine quail were randomized to one of four treatments for lowering cholesterol.

**Usage**  
`data(quail)`

**Format**  
A data frame with 39 observations on the following 2 variables.

- **treat** a factor with levels 1 2 3 4
- **ldl** a numeric vector

**Source**  

**Examples**  
```r
data(quail)
boxplot(ldl~treat,data=quail)
```

---

**raov**  
*R ANOVA*

**Description**  
Returns full model fit and robust ANOVA table for all main effects and interactions.

**Usage**  
`raov(f, data = list(), ...)`

**Arguments**

<table>
<thead>
<tr>
<th>f</th>
<th>an object of class formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>an optional data frame</td>
</tr>
<tr>
<td>...</td>
<td>additional arguments</td>
</tr>
</tbody>
</table>
Details

Based on reduction in dispersion tests for testing main effects and interaction. Uses an algorithm described in Hocking (1985).

Value

- `table`: Description of 'comp1'
- `fit`: full model fit returned from rfit
- `residuals`: the residuals, i.e. y-yhat
- `fitted.values`: yhat = x betahat
- `call`: Call to the function

Author(s)

Joseph McKean, John Kloke

References


See Also

- `rfit`, `oneway.rfit`

Examples

```r
raov(logSurv~Poison+Treatment,data=BoxCox)
```

---

**rfit**

**Rank-based Estimates of Regression Coefficients**

Description

Minimizes Jaeckel’s dispersion function to obtain a rank-based solution for linear models.

Usage

```r
rfit(formula, data = list(), ...)
```

# Default S3 method:
rfit(formula, data, subset, yhat0 = NULL,
     scores = Rfit::wscores, symmetric = FALSE, TAU = "F0", ...)

Arguments

formula an object of class formula
data an optional data frame
subset an optional argument specifying the subset of observations to be used
yhat0 an n by vector of initial fitted values, default is NULL
scores an object of class 'scores'
symmetric logical. If 'FALSE' uses median of residuals as estimate of intercept
TAU version of estimation routine for scale parameter. F0 for Fortran, R for (slower)
R, N for none
... additional arguments to be passed to fitting routines

Details

Rank-based estimation involves replacing the L2 norm of least squares estimation with a pseudo-norm which is a function of the ranks of the residuals. That is, in rank estimation, the usual notion of Euclidean distance is replaced with another measure of distance which is referred to as Jaeckel’s (1972) dispersion function. Jaeckel’s dispersion function depends on a score function and a library of commonly used score functions is included. e.g. linear (Wilcoxon) and normal (Gaussian) scores. If an initial fit is not supplied (i.e. yhat0 = NULL) then initial fit is based on a LS fit.

Value

coefficients estimated regression coefficients with intercept
residuals the residuals, i.e. y-yhat
fitted.values yhat = x betahat
xc centered design matrix
tauhat estimated value of the scale parameter tau
taushat estimated value of the scale parameter tau_s
betahat estimated regression coefficients
call Call to the function

Author(s)

John Kloke, Joseph McKea

References


rstudent.rfit

See Also

summary.rfit drop.test rstudent.rfit

Examples

data(baseball)
data(wscores)
fit<-rfit(weight~height,data=baseball)
summary(fit)

### set the starting value
x1 <- runif(47); x2 <- runif(47); y <- 1 + 0.5*x1 + rnorm(47)
# based on a fit to a sub-model
rfit(y~x1+x2,yhat=fitted.values(rfit(y~x1)))

---

rstudent.rfit  

Studentized Residuals for Rank-Based Regression

Description

Returns the Studentized residuals based on rank-based estimation.

Usage

## S3 method for class 'rfit'
rstudent(model,...)

Arguments

model  
an object of class rfit

...  
additional arguments. currently not used.

Author(s)

John Kloke, Joseph McKean

References


See Also

rfit
**Examples**

```r
x <- runif(47)
y <- rcauchy(47)
qqnorm(rstudent(fit <- rfit(y ~ x)))
plot(x, rstudent(fit)); abline(h = c(-2, 2))
```

---

**scores-class**

Class "scores"

**Description**

A score function and it’s corresponding derivative is required for rank-based estimation. This object puts them together.

**Objects from the Class**

Objects can be created by calls of the form `new("scores", ...)`. 

**Slots**

- `phi`: Object of class "function" the score function
- `Dphi`: Object of class "function" the first derivative of the score function
- `param`: Object of class "param"

**Author(s)**

John Kloke

**References**


**See Also**

`param`

**Examples**

`showClass("scores")`
**serumLH**  

*Serum Level of luteinizing hormone (LH)*

**Description**

Hollander and Wolfe (1999) discuss a 2 by 5 factorial design for a study to determine the effect of light on the release of luteinizing hormone (LH). The factors in the design are: light regimes at two levels (constant light and 14 hours of light followed by 10 hours of darkness) and a luteinizing release factor (LRF) at 5 different dosage levels. The response is the level of luteinizing hormone (LH), nanograms per ml of serum in blood samples. Sixty rats were put on test under these 10 treatment combinations, six rats per combination.

**Usage**

```r
data(serumLH)
```

**Format**

A data frame with 60 observations on the following 3 variables.

- **serum** a numeric vector
- **light.regime** a factor with levels *Constant Intermittent*
- **LRF.dose** a factor with levels 0 10 1250 250 50

**Source**


**References**


**Examples**

```r
data(serumLH)
raov(serum~light.regime + LRF.dose + light.regime*LRF.dose, data = serumLH)
```
signedrank  

Signed-Rank Estimate of Location (Intercept)

Description

Returns the signed-rank estimate of intercept with is equivalent to the Hodges-Lehmann estimate of the residuals.

Usage

signedrank(x)

Arguments

x numeric vector

Value

Returns the median of the Walsh averages.

Author(s)

John Kloke, Joseph McKean

References


See Also

walsh

Examples

## The function is currently defined as
function (x)
median(walsh(x))
**summary.oneway.rfit**  
*Provides a summary for the oneway anova based on an R fit.*

### Description

Provides a summary for the oneway anova based on an R fit including a test for main effects as tests for pairwise comparisons.

### Usage

```
## S3 method for class 'oneway.rfit'
summary(object, alpha=0.05, method=confintadjust.methods,...)
```

### Arguments

- **object**  
an object of class 'oneway.rfit', usually, a result of a call to 'oneway.rfit'
- **alpha**  
Experimentwise Error Rate
- **method**  
method used in confidence interval adjustment
- **...**  
additional arguments

### Author(s)

John Kloke, Joseph McKean

### References


### Examples

```
data(quail)
oneway.rfit(quail$ldl, quail$treat)
```

---

**summary.rfit**  
*Summarize Rank-Based Linear Model Fits*

### Description

Provides a summary similar to the traditional least squares fit.

### Usage

```
## S3 method for class 'rfit'
summary(object, overall.test,...)
```
Arguments

object: an object of class 'rfit', usually, a result of a call to 'rfit'
overall.test: either 'wald' or 'drop'
...: additional arguments

Details

Provides summary statistics based on a rank-based fit. A table of estimates, standard errors, t-ratios, and p-values are provided. An overall test of the explanatory variables is provided; the default is to use a Wald test. A drop in dispersion test is also available in which case a robust R^2 is provided as well.

Author(s)

John Kloke

References


Examples

data(baseball)
fit<-rfit(weight~height,data=baseball)
summary(fit)
summary(fit,overall.test='drop')

---

taufuncs

Internal Functions for Estimating tau

Description

These are internal functions used for calculating the scale parameter tau necessary for estimating the standard errors of coefficients for rank-regression.

Usage

hstarreadyscr(ehat,asc,ascpr)
hstar(abdord, wtord, const, n, y)
looptau(delta, abdord, wtord, const, n)
pairup(x,type="less")
Arguments

ehat          Full model residuals
delta         Window parameter (proportion) used in the Koul et al. estimator of tau. Default value is 0.80. If the ratio of sample size to number of regression parameters (n to p) is less than 5, larger values such as 0.90 to 0.95 are more appropriate.
y             Argument of function hstar
abbrd         Ordered absolute differences of residuals
wtdrd         Standardized (by const) ordered absolute differences of residuals
const         Range of score function
n             Sample size
x             Argument for pairup
type          Argument for the function pairup
asc           scores
ascpr         derivative of the scores

Author(s)

Joseph McKean, John Kloke

References


See Also

gettau, rfit

taustar          Estimate of the Scale Parameter taustar

Description

An estimate of the scale parameter taustar = 1/(2*f(0)) is needed for the standard error of the intercept in rank-based regression.

Usage

taustar(e, p, conf = 0.95)
Arguments

- `e` : n x 1 vector of full model residuals
- `p` : is the number of regression coefficients (without the intercept)
- `conf` : confidence level of CI used

Details

Confidence interval estimate of \( \tau_{\text{star}} \). See, for example, Hettmansperger and McKean (1998) p.7-8 and p.25-26.

Value

Length-one numeric object containing the estimated scale parameter \( \tau_{\text{star}} \).

Author(s)

Joseph McKean, John Kloke

References


See Also

`rfit`

Examples

```r
## This is an internal function. See rfit for user-level examples.
```

---

**telephone**

*Telephone Data*

Description

The number of telephone calls (in tens of millions) made in Belgium from 1950-1973.

Usage

`data(telephone)`

Format

A data frame with 24 observations on the following 2 variables.

- `year` : years since 1950 AD
- `calls` : number of telephone calls in tens of millions
vcov.rfit

Source

References

Examples
data(telephone)
plot(telephone)
abline(rfit(calls~year,data=telephone))

---

vcov.rfit  
Variance-Covariance Matrix for Rank-Based Regression

Description
Returns the variance-covariance matrix of the regression estimates from an object of type rfit.

Usage
```r
## S3 method for class 'rfit'
vcov(object, intercept = NULL,...)
```

Arguments
- `object` an object of type rfit
- `intercept` logical. If TRUE include the variance-covariance estimates corresponding to the intercept
- `...` additional arguments

Author(s)
John Kloke

References

See Also
rfit
wald.test.overall  

**Overall Wald test**

**Description**

Conducts a Wald test of all regression parameters are zero

**Usage**

```r
wald.test.overall(fit)
```

**Arguments**

- `fit`  
  result from a rfit

**Author(s)**

John Kloke

**References**


**Examples**

```r
x <- rnorm(47)
y <- rnorm(47)
wald.test.overall(rfit(y~x))
```

walsh  

**Walsh Averages**

**Description**

Given a list of n numbers, the Walsh averages are the *latex* pairwise averages.

**Usage**

```r
walsh(x)
```

**Arguments**

- `x`  
  A numeric vector
walsh

Value
The Walsh averages.

Author(s)
John Kloke, Joseph McKeane

References

See Also
signedrank

Examples

median(walsh(rnorm(100))) # Hodges-Lehmann estimate of location

## The function is currently defined as
function (x)
{
  n <- length(x)
  w <- vector(n * (n + 1)/2, mode = "numeric")
  ind <- 0
  for (i in 1:n) {
    for (j in i:n) {
      ind <- ind + 1
      w[ind] <- 0.5 * (x[i] + x[j])
    }
  }
  return(w)
}
Index

* classes
  param-class, 18
  scores-class, 24
* datasets
  allscores, 3
  baseball, 4
  bbsalaries, 5
  BoxCox, 6
  CardioRiskFactors, 7
  ffa, 11
  quail, 20
  serumLH, 25
  telephone, 30
* design
  oneway.rfit, 17
  raov, 20
* htest
  drop.test, 10
  wald.test.overall, 32
* methods
  getScores-methods, 12
  getScoresDeriv-methods, 12
* nonparametric
  oneway.rfit, 17
  raov, 20
  rfit, 21
  Rfit-package, 2
* package
  Rfit-package, 2
* regression
  rfit, 21
  Rfit-package, 2
  wald.test.overall, 32
* robust
  drop.test, 10
  oneway.rfit, 17
  raov, 20
  rfit, 21
  Rfit-package, 2
  rstudent.rfit, 23
  wald.test.overall, 32
  allscores, 3
  baseball, 4
  bbsalaries, 5
  bentscores1(allscores), 3
  bentscores2(allscores), 3
  bentscores3(allscores), 3
  bentscores4(allscores), 3
  BoxCox, 6
  CardioRiskFactors, 7
cellx (kwayr), 16
confintadjust, 8
disp, 9, 14
drop.test, 9, 10, 19, 23
ffa, 11
getScores(getScores-methods), 12
getScores, scores-method
  (getScores-methods), 12
getScores-methods, 12
getScoresDeriv
  (getScoresDeriv-methods), 12
getScoresDeriv, scores-method
  (getScoresDeriv-methods), 12
getScoresDeriv-methods, 12
gettau, 13, 29
gettauF0(gettau), 13
grad, 14
hstar (taufuncs), 28
hstarreadyscr (taufuncs), 28
jaeckel, 15
khmat (kwayr), 16
INDEX

kwayr, 16
logGFscores (allscores), 3
logrank.scores (allscores), 3
looptau (taufuncs), 28
nscale (taufuncs), 28
nscores (allscores), 3
oneway.rfit, 9, 17, 21
optim, 16
pairup (taufuncs), 28
param, 24
param (param-class), 18
param-class, 18
pasteColsRfit (kwayr), 16
print.drop.test (print.rfit), 19
print.oneway.rfit (print.rfit), 19
print.raov (print.rfit), 19
print.rfit, 19
print.summary.oneway.rfit (print.rfit), 19
print.summary.rfit (print.rfit), 19
quail, 20
raov, 17, 20
redmod (kwayr), 16
Rfit (Rfit-package), 2
rfit, 9, 11, 12, 14, 16, 18, 19, 21, 23, 29–31
Rfit-package, 2
rstudent.rfit, 23, 23
scores, 19
scores (scores-class), 24
scores-class, 24
serumLH, 25
signedrank, 26, 33
subsets (kwayr), 16
summary.oneway.rfit, 27
summary.rfit, 9, 19, 23, 27
taufuncs, 28
taustar, 29
telephone, 30
vcov.rfit, 31
wald.test.overall, 32
walsh, 26, 32
wscores (allscores), 3