

# Package ‘Rfit’

May 7, 2026

**Type** Package

**Title** Rank-Based Estimation for Linear Models

**Version** 0.27.0

**Date** 2024-05-25

**Author** John Kloke, Joseph McKean

**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** 2024-05-25 17:50:06 UTC

## Contents

|                             |    |
|-----------------------------|----|
| 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 |

|                                  |           |
|----------------------------------|-----------|
| getScoresDeriv-methods . . . . . | 12        |
| gettau . . . . .                 | 13        |
| grad . . . . .                   | 14        |
| jaeckel . . . . .                | 15        |
| kwayr . . . . .                  | 17        |
| oneway.rfit . . . . .            | 18        |
| param-class . . . . .            | 19        |
| print.rfit . . . . .             | 20        |
| quail . . . . .                  | 20        |
| raov . . . . .                   | 21        |
| rfit . . . . .                   | 22        |
| rstudent.rfit . . . . .          | 24        |
| scores-class . . . . .           | 25        |
| serumLH . . . . .                | 25        |
| signedrank . . . . .             | 26        |
| summary.oneway.rfit . . . . .    | 27        |
| summary.rfit . . . . .           | 28        |
| taufuncs . . . . .               | 29        |
| taustar . . . . .                | 30        |
| telephone . . . . .              | 31        |
| vcov.rfit . . . . .              | 31        |
| wald.test.overall . . . . .      | 32        |
| walsh . . . . .                  | 33        |
| <b>Index</b>                     | <b>35</b> |

---

Rfit-package

*Rank-Based Estimates and Inference for Linear Models*


---

## Description

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

## Details

|           |                          |
|-----------|--------------------------|
| Package:  | Rfit                     |
| Type:     | Package                  |
| Version:  | 0.27.0                   |
| Date:     | 2024-05-25               |
| License:  | GPL (version 2 or later) |
| LazyLoad: | yes                      |

**Author(s)**

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

**References**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

Jaeckel, L. A. (1972). Estimating regression coefficients by minimizing the dispersion of residuals. *Annals of Mathematical Statistics*, 43, 1449 - 1458.

Jureckova, J. (1971). Nonparametric estimate of regression coefficients. *Annals of Mathematical Statistics*, 42, 1328 - 1338.

**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

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

## Examples

```
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

```
data(baseball)
```

**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**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

**Examples**

```
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**

```
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

**Source**

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

**References**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

**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.

logSurv log Survival Time

Poison a factor indicating poison level

Treatment a factor indicating treatment level

**Source**

Box, G.E.P. and Cox, D.R. (1964), An analysis of transformations, *Journal of the Royal Statistical Society, Series B, Methodological*, 26, 211-252.

**References**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

**Examples**

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

---

|                   |                                    |
|-------------------|------------------------------------|
| CardioRiskFactors | <i>Cardiovascular risk factors</i> |
|-------------------|------------------------------------|

---

**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**

```
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 lipoprotien(?)  
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**

Data set and description taken from Heritier et. al. (2009) (c.f. Conen et. al. 2004).

**Source**

Heritier, S., Cantoni, E., Copt, S., and Victoria-Feser, M. (2009), *Robust Methods in Biostatistics*, New York: John Wiley and Sons.

Conen, D., Wietlisbach, V., Bovet, P., Shamlaye, C., Riesen, W., Paccaud, F., and Burnier, M. (2004), Prevalence of hyperuricemia and relation of serum uric acid with cardiovascular risk factors in a developing country. *BMC Public Health*.

**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 | <i>Confidence interval adjustment methods</i> |
|---------------|---|

---

**Description**

Returns the critical value to be used in calculating adjusted confidence intervals. Currently provides methods for Boneferroni 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**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

**See Also**[oneway.rfit](#)

---

|      |                                     |
|------|-------------------------------------|
| disp | <i>Jaekel's Dispersion Function</i> |
|------|-------------------------------------|

---

**Description**

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

**Author(s)**

John Kloke, Joseph McKean

**References**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

Jaekel, L. A. (1972). Estimating regression coefficients by minimizing the dispersion of residuals. *Annals of Mathematical Statistics*, 43, 1449 - 1458.

**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**

```
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$disp > fitR$disp`. 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**

|                      |   |
|----------------------|---|
| <code>F</code>       | Value of the F test statistic                                     |
| <code>p.value</code> | The observed significance level of the test (using an F quantile) |
| <code>RD</code>      | Reduced model dispersion minus Full model dispersion              |
| <code>tauhat</code>  | Estimate of the scale parameter (using the full model residuals)  |
| <code>df1</code>     | numerator degrees of freedom                                      |
| <code>df2</code>     | denominator degrees of freedom                                    |

**Author(s)**

John Kloke, Joseph McKean

## References

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

## See Also

[rfit](#)

## Examples

```
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

```
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

Morrison, D.F. (1983), *Applied Linear Statistical Models*, Englewood Cliffs, NJ:Prentice Hall.

**References**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

**Examples**

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

---

getScoresDeriv-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](#)

---

|        |  |
|--------|--|
| gettau | <i>Estimate of the scale parameter tau</i> |
|--------|--|

---

### Description

An estimate of the scale parameter tau may be used for the standard errors of the coefficients in rank-based regression.

### Usage

```
gettau(ehat, p, scores = Rfit::wscores, delta = 0.8, hparm = 2, ...)
gettauF0(ehat, p, scores = Rfit::wscores, delta = 0.8, hparm = 2, ...)
```

### Arguments

|        |  |
|--------|--|
| ehat   | vector of length n: full model residuals   |
| p      | scalar: number of regression coefficients (excluding the intercept); see Details |
| scores | object of class scores, defaults to Wilcoxon scores                              |
| delta  | confidence level; see Details  |
| hparm  | used in Huber's degrees of freedom correction; see Details                       |
| ...    | additional arguments. currently unused   |

### Details

For rank-based analyses of linear models, the estimator  $\hat{\tau}$  of the scale parameter  $\tau$  plays a standardizing role in the standard errors (SE) of the rank-based estimators of the regression coefficients and in the denominator of Wald-type and the drop-in-dispersion test statistics of linear hypotheses. `rfit` currently implements the KSM (Koul, Sievers, and McKean 1987) estimator of tau.

The functions `gettau` and `gettauF0` are both available to compute the KSM estimate and may be called from `rfit` and used for inference. The default is to use the faster FORTRAN version `gettauF0` via the option `TAU='F0'`. The R version, `gettau`, may be much slower especially when sample sizes are large; this version may be called from `rfit` using the option `TAU='R'`.

The KSM estimator `tauhat` is a density type estimator that has the bandwidth given by  $t_\delta / \sqrt{rtn}$ , where  $t_\delta$  is the  $\delta - th$  quantile of the cdf  $H(y)$  given in expression (3.7.2) of Hettmansperger and McKean (2011), with the corresponding estimator  $\hat{H}$ , given in expression (3.7.7) of Hettmansperger and McKean (2011).

Based on simulation studies, most situations where  $(n/p \geq 6)$ , the default `delta = 0.80` provides a valid rank-based analysis (McKean and Sheather, 1991). For situations with  $n/p < 6$ , caution is needed as the KSM estimate is sensitive to choice of bandwidth. McKean and Sheather (1991) recommend using a value of 0.95 for `delta` in such situations.

To correct for heavy-tailed random errors, Huber (1973) proposed a degree of freedom correction for the M-estimate scale parameter. The correction is given by  $K = 1 + [p * (1 - h_c) / n * h_c]$  where  $h_c$  is the proportion of standardized residuals in absolute value less than the parameter `hparm`. This correction  $K$  is used as a multiplicative factor to `tauhat`. The default value of `hparm` is set at 2.

The usual degrees of freedom correction,  $\sqrt{n/(n-p)}$ , is also used as a multiplicative factor to `tauhat`.

**Value**

Length one numeric object.

**Author(s)**

Joseph McKean, John Kloke

**References**

- Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.
- Huber, P.J. (1973), Robust regression: Asymptotics, conjectures and Monte Carlo, *Annals of Statistics*, 1, 799–821.
- Koul, H.L., Sievers, G.L., and McKean, J.W. (1987), An estimator of the scale parameter for the rank analysis of linear models under general score functions, *Scandinavian Journal of Statistics*, 14, 131–141.
- McKean, J. W. and Sheather, S. J. (1991), Small Sample Properties of Robust Analyses of Linear Models Based on R-Estimates: A Survey, in *Directions in Robust Statistics and Diagnostics, Part II*, Editors: W.\ Stahel and S.\ Weisberg, Springer-Verlag: New York, 1–19.

**See Also**

[rfit](#)

**Examples**

```
# For a standard normal distribution the parameter tau has the value 1.023327 (sqrt(pi/3)).
set.seed(283643659)
n <- 12; p <- 6; y <- rnorm(n); x <- matrix(rnorm(n*p),ncol=p)
tau1 <- rfit(y~x)$tauhat; tau2 <- rfit(y~x,delta=0.95)$tauhat
c(tau1,tau2) # 0.5516708 1.0138415
n <- 120; p <- 6; y <- rnorm(n); x <- matrix(rnorm(n*p),ncol=p)
tau3 <- rfit(y~x)$tauhat; tau4 <- rfit(y~x,delta=0.95)$tauhat
c(tau3,tau4) # 1.053974 1.041783
```

---

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**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

Jaeckel, L. A. (1972). Estimating regression coefficients by minimizing the dispersion of residuals. *Annals of Mathematical Statistics*, 43, 1449 - 1458.

Jureckova, J. (1971). Nonparametric estimate of regression coefficients. *Annals of Mathematical Statistics*, 42, 1328 - 1338.

**See Also**

[disp](#)

**Examples**

```
## 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 with respect to beta.

**Usage**

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

**Arguments**

|                      |  |
|----------------------|--|
| <code>x</code>       | n by p design matrix                                 |
| <code>y</code>       | n by 1 response vector                               |
| <code>beta0</code>   | initial estimate of beta                             |
| <code>scores</code>  | object of class 'scores'                             |
| <code>control</code> | control passed to fitting routine                    |
| <code>...</code>     | additional arguments to be passed to fitting routine |

**Details**

Jaeckel's dispersion function (Jaeckel 1972) is a convex function which measures the distance between the observed responses  $y$  and the fitted values  $x\beta$ . The dispersion function is a sum of the products of the residuals,  $y - x\beta$ , and the scored ranks of the residuals. A rank-based fit minimizes the dispersion function; see McKean and Schrader (1980) and Kloke and McKean (2012) for discussion. `jaeckel` uses `optim` with the 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^(3/4)` and maximum number of iterations is set to 200.

`jaeckel` is intended to be an internal function. See `rfit` for a general purpose function.

**Value**

Results of `optim` are returned.

**Author(s)**

John Kloke

**References**

- Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.
- Jaeckel, L. A. (1972), Estimating regression coefficients by minimizing the dispersion of residuals. *Annals of Mathematical Statistics*, 43, 1449 - 1458.
- Kapenga, J. A., McKean, J. W., and Vidmar, T. J. (1988), *RGLM: Users Manual*, Statist. Assoc. Short Course on Robust Statistical Procedures for the Analysis of Linear and Nonlinear Models, New Orleans.

**See Also**

[optim](#), [rfit](#)

**Examples**

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

---

kwayr

*Internal Functions for K-Way analysis of variance*


---

**Description**

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

**Usage**

```
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     | Seperator used in pasteColsRfit                                       |
| ...     | additional arguments  |

**Note**

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

**Author(s)**

Joseph McKean, John Kloke

**References**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

Hocking, R. R. (1985), *The Analysis of Linear Models*, Monterey, California: Brooks/Cole.

**See Also**

[raov](#)

---

oneway.rfit

*Rank-based Oneway Analysis of Variance*

---

**Description**

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

**Usage**

```
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 |
| ...      | additional arguments                                    |

**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**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

**See Also**

[rfit](#)

**Examples**

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

---

|             |                      |
|-------------|----------------------|
| param-class | <i>Class "param"</i> |
|-------------|----------------------|

---

**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 for 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**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

**Examples**

```
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**

|      |                            |
|------|----------------------------|
| f    | an object of class formula |
| data | an optional data frame     |
| ...  | additional arguments       |

**Details**

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

**Value**

|               |                                   |
|---------------|-----------------------------------|
| table         | Description of 'compl'            |
| 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**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.  
 Hocking, R. R. (1985), *The Analysis of Linear Models*, Monterey, California: Brooks/Cole.

**See Also**

[rfit](#), [oneway.rfit](#)

**Examples**

```
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**

```
rfit(formula, data = list(), ...)
## Default S3 method:
rfit(formula, data, subset, yhat0 = NULL,
      scores = Rfit::wscores, symmetric = FALSE, TAU = "F0",
      betahat0 = NULL, ...)
```

**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 1 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 |
| betahat0  | a p by 1 vector of initial parameter estimates, default is NULL                                 |
| ...       | 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 residuals and the scored ranks of the residuals. That is, in rank-based 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; eg., linear (Wilcoxon) and normal (Gaussian) scores. If an initial fit is not supplied (i.e. `yhat0 = NULL` and `betahat0 = NULL`) then initial fit is based on a LS fit.

Estimation of scale parameter `tau` is provided which may be used for inference.

## Value

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

## Author(s)

John Kloke, Joseph McKean

## References

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

Jaeckel, L. A. (1972). Estimating regression coefficients by minimizing the dispersion of residuals. *Annals of Mathematical Statistics*, 43, 1449 - 1458.

Jureckova, J. (1971). Nonparametric estimate of regression coefficients. *Annals of Mathematical Statistics*, 42, 1328 - 1338.

## 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,yhat0=fitted.values(rfit(y~x1)))

### set value of delta used in estimation of tau ###
w <- factor(rep(1:3,each=3))
y <- rt(9,9)
rfit(y~w)$tauhat
rfit(y~w,delta=0.95)$tauhat # recommended when n/p < 5
```

---

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

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

### See Also

[rfit](#)

### Examples

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

---

|              |                       |
|--------------|-----------------------|
| scores-class | <i>Class "scores"</i> |
|--------------|-----------------------|

---

**Description**

A score function and its 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**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

**See Also**

[param](#)

**Examples**

```
showClass("scores")
```

---

|         |  |
|---------|--|
| serumLH | <i>Serum Level of luteinizing hormone (LH)</i> |
|---------|--|

---

**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**

```
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**

Hollander, M. and Wolfe, D.A. (1999), *Nonparametric Statistical Methods*, New York: Wiley.

**References**

Hollander, M. and Wolfe, D.A. (1999), *Nonparametric Statistical Methods*, New York: Wiley.

**Examples**

```
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**

- Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.
- Hollander, M. and Wolfe, D.A. (1999), *Nonparametric Statistical Methods*, New York: Wiley.

**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**

- Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

**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

|                           |  |
|---------------------------|--|
| <code>object</code>       | an object of class 'rfit', usually, a result of a call to 'rfit' |
| <code>overall.test</code> | either 'wald' or 'drop'  |
| <code>...</code>          | 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

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

### 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  |
| abdord | Ordered absolute differences of residuals   |
| wtord  | 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**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

Koul, H.L., Sievers, G.L., and McKean, J.W. (1987) An estimator of the scale parameter for the rank analysis of linear models under general score functions, *Scandinavian Journal of Statistics*, 14, 131-141.

**See Also**

[gettau](#), [rfit](#)

---

`taustar`*Estimate of the Scale Parameter taustar*

---

**Description**

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

**Usage**

```
taustar(e, p, conf = 0.95)
```

**Arguments**

|                   |  |
|-------------------|--|
| <code>e</code>    | n x 1 vector of full model residuals                             |
| <code>p</code>    | is the number of regression coefficients (without the intercept) |
| <code>conf</code> | confidence level of CI used                                      |

**Details**

Confidence interval estimate of taustar. See, for example, Hettmansperger and McKean (1998) p.7-8 and p.25-26.

**Value**

Length-one numeric object containing the estimated scale parameter taustar.

**Author(s)**

Joseph McKean, John Kloke

**References**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

**See Also**

[rfit](#)

**Examples**

```
## 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

### Source

Rousseeuw, P.J. and Leroy, A.M. (1987), *Robust Regression and Outlier Detection*, New York: Wiley.

### References

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

### 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

```
## 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**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

**See Also**

[rfit](#)

---

|                   |                          |
|-------------------|--------------------------|
| wald.test.overall | <i>Overall Wald test</i> |
|-------------------|--------------------------|

---

**Description**

Conducts a Wald test of all regression parameters are zero

**Usage**

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

**Arguments**

|     |                    |
|-----|--------------------|
| fit | result from a rfit |
|-----|--------------------|

**Author(s)**

John Kloke

**References**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

**Examples**

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

---

|       |                       |
|-------|-----------------------|
| walsh | <i>Walsh Averages</i> |
|-------|-----------------------|

---

**Description**

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

**Usage**

```
walsh(x)
```

**Arguments**

`x`                    A numeric vector

**Value**

The Walsh averages.

**Author(s)**

John Kloke, Joseph McKean

**References**

Hettmansperger, T.P. and McKean J.W. (2011), *Robust Nonparametric Statistical Methods, 2nd ed.*, New York: Chapman-Hall.

Hollander, M. and Wolfe, D.A. (1999), *Nonparametric Statistical Methods*, New York: Wiley.

**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, 19
  - scores-class, 25
- \* **datasets**
  - allscores, 3
  - baseball, 4
  - bbsalaries, 5
  - BoxCox, 6
  - CardioRiskFactors, 7
  - ffa, 11
  - quail, 20
  - serumLH, 25
  - telephone, 31
- \* **design**
  - oneway.rfit, 18
  - raov, 21
- \* **htest**
  - drop.test, 10
  - wald.test.overall, 32
- \* **methods**
  - getScores-methods, 12
  - getScoresDeriv-methods, 12
- \* **nonparametric**
  - oneway.rfit, 18
  - raov, 21
  - rfit, 22
  - Rfit-package, 2
- \* **package**
  - Rfit-package, 2
- \* **regression**
  - rfit, 22
  - Rfit-package, 2
  - wald.test.overall, 32
- \* **robust**
  - drop.test, 10
  - oneway.rfit, 18
  - raov, 21
  - rfit, 22
  - Rfit-package, 2
  - rstudent.rfit, 24
  - 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), 17
- confintadjust, 8
- disp, 9, 15
- drop.test, 9, 10, 20, 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), 29
- hstarreadyscr (taufuncs), 29
- jaeckel, 15
- khmat (kwayr), 17

kwayr, 17

logGFscores (allscores), 3  
logrank.scores (allscores), 3  
looptau (taufuncs), 29

nscale (taufuncs), 29  
nscores (allscores), 3

oneway.rfit, 9, 18, 22  
optim, 16

pairup (taufuncs), 29  
param, 25  
param (param-class), 19  
param-class, 19  
pasteColsRfit (kwayr), 17  
print.drop.test (print.rfit), 20  
print.oneway.rfit (print.rfit), 20  
print.raov (print.rfit), 20  
print.rfit, 20  
print.summary.oneway.rfit (print.rfit),  
20  
print.summary.rfit (print.rfit), 20

quail, 20

raov, 18, 21  
redmod (kwayr), 17  
Rfit (Rfit-package), 2  
rfit, 9, 11, 12, 14, 16, 19, 20, 22, 22, 24, 29,  
30, 32  
Rfit-package, 2  
rstudent.rfit, 23, 24

scores, 19  
scores (scores-class), 25  
scores-class, 25  
serumLH, 25  
signedrank, 26, 33  
subsets (kwayr), 17  
summary.oneway.rfit, 27  
summary.rfit, 9, 20, 23, 28

taufuncs, 29  
taustar, 30  
telephone, 31

vcov.rfit, 31

wald.test.overall, 32  
walsh, 27, 33  
wscores (allscores), 3