

# Package ‘contrast’

February 19, 2015

**Version** 0.19

**Date** 2013-11-15

**Title** A collection of contrast methods

**Author** Max Kuhn, contributions from Steve Weston, Jed Wing,  
James Forester and Thorn Thaler

**Maintainer** Max Kuhn <max.kuhn@pfizer.com>

**Depends** R (>= 2.2.1), rms

**Suggests** lattice, nlme, Hmisc, geepack, MASS, sandwich

**Description** Contrast methods, in the style of the Design package,  
for fit objects produced by the lm, glm, gls, and geese functions.

**License** GPL-2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2013-11-16 06:37:04

## R topics documented:

contrast.lm . . . . . 1

**Index** 4

---

contrast.lm                      *General Contrasts of Regression Coefficients*

---

## Description

This function computes one or more contrasts of the estimated regression coefficients in a fit from one of the functions in Design, along with standard errors, confidence limits, t or Z statistics, P-values.

**Usage**

```

contrast(fit, ...)
## S3 method for class 'lm'
contrast(fit, ...)
## S3 method for class 'gls'
contrast(fit, ...)
## S3 method for class 'lme'
contrast(fit, ...)
## S3 method for class 'geese'
contrast(fit, ...)

## S3 method for class 'contrast'
print(x, X=FALSE, fun=function(u)u, ...)

```

**Arguments**

<code>fit</code>	a fit of class "lm", "glm", etc.
<code>...</code>	arguments to pass to the computational code. The arguments are listed in the Details section below.
<code>x</code>	result of contrast
<code>X</code>	set X=TRUE to print design matrix used in computing the contrasts (or the average contrast)
<code>fun</code>	a function to transform the contrast, SE, and lower and upper confidence limits before printing. For example, specify fun=exp to anti-log them for logistic models.

**Details**

These functions mirror `contrast.rms`.

There are some between-package inconsistencies regarding degrees of freedom in some models. See the package vignette for more details.

Fold changes are calculated for each hypothesis. When `fcType = "simple"`, the ratio of the a group predictions over the b group predictions are used. When `fcType = "signed"`, the ratio is used if it is greater than 1; otherwise the negative inverse (e.g.,  $-1/\text{ratio}$ ) is returned.

Arguments to the contrast functions are: `a`: a list containing settings for all predictors that you do not wish to set to default (adjust-to) values. Usually you will specify two variables in this list, one set to a constant and one to a sequence of values, to obtain contrasts for the sequence of values of an interacting factor. The `gendata` function will generate the necessary combinations and default values for unspecified predictors.

`a`: another list that generates the same number of observations as `a`, unless one of the two lists generates only one observation. In that case, the design matrix generated from the shorter list will have its rows replicated so that the contrasts assess several differences against the one set of predictor values. This is useful for comparing multiple treatments with control, for example. If `b` is missing, the design matrix generated from `a` is analyzed alone.

`covType`: a string matching the method for estimating the covariance matrix. The default value produces the typical estimate. See [vcovHC](#) for options.

`cnames`: vector of character strings naming the contrasts when `type="individual"`. Usually `cnames` is not necessary as `contrast.rms` tries to name the contrasts by examining which predictors are varying consistently in the two lists. `cnames` will be needed when you contrast "non-comparable" settings, e.g., you compare `list(treat="drug", age=c(20,30))` with `list(treat="placebo"), age=c(40,30)`.

`type`: set `type="average"` to average the individual contrasts (e.g., to obtain a Type II or III contrast)

`weights`: a numeric vector, used when `type="average"`, to obtain weighted contrasts

`conf.int`: confidence level for confidence intervals for the contrasts

`env`: environment in which evaluate fit

`fcFun`: a function to transform the numerator and denominator of fold changes

`fcType`: a character string: "simple", "log" or "signed"

### Value

a list of class "contrast.Design" containing the elements `Contrast`, `SE`, `Z`, `var`, `df.residual`, `Lower`, `Upper`, `Pvalue`, `X`, `cnames`, which denote the contrast estimates, standard errors, Z or t-statistics, variance matrix, residual degrees of freedom (this is NULL if the model was not ols), lower and upper confidence limits, 2-sided P-value, design matrix, and contrast names (or NULL). Also, an element called `foldChange`.

### See Also

[contrast.rms](#), [vcovHC](#)

### Examples

```
library(nlme)
Orthodont2 <- Orthodont
Orthodont2$newAge <- Orthodont$age - 11
fm1Orth.lme2 <- lme(distance ~ Sex*newAge, data = Orthodont2, random = ~ newAge | Subject)
summary(fm1Orth.lme2)

contrast(fm1Orth.lme2,
         a = list(Sex = levels(Orthodont2$Sex), newAge = 8 - 11),
         b = list(Sex = levels(Orthodont2$Sex), newAge = 10 - 11))
```

# Index

\*Topic **models**

contrast.lm, 1

\*Topic **regression**

contrast.lm, 1

contrast (contrast.lm), 1

contrast.lm, 1

contrast.rms, 3

print.contrast (contrast.lm), 1

vcovHC, 2, 3