Package 'contrast'

October 25, 2026

Title A Collection of Contrast Methods

Version 0.25.0

Date 2026-10-24

Description One degree of freedom contrasts for 'lm', 'glm', 'gls', and 'geese' objects.

License GPL-2

URL https://github.com/alanocallaghan/contrast

BugReports https://github.com/alanocallaghan/contrast/issues

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

Imports nlme, sandwich, rms

Suggests knitr, kableExtra, dplyr, ggplot2, tidyr, rmarkdown, testthat, covr, geepack, splines

VignetteBuilder knitr

Language en-US

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Alan O'Callaghan [cre], Max Kuhn [aut], Steve Weston [aut], Jed Wing [aut], James Forester [aut], Thorn Thaler [aut]

Maintainer Alan O'Callaghan <alan.ocallaghan@outlook.com>

Repository CRAN

Date/Publication 2026-10-24 23:30:08 UTC

Contents

contrast.lm 2
print.contrast 4
two_factor_crossed 5
two_factor_incompl 6

Index **7**

`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

```
## 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, ...)

contrast_calc(
  fit,
  a,
  b,
  cnames = NULL,
  type = c("individual", "average"),
  weights = "equal",
  conf.int = 0.95,
  fcType = "simple",
  fcFunc = I,
  covType = NULL,
  ...,
  env = parent.frame(2)
)
```

Arguments

fit	A fit of class lm, glm, etc.
...	For contrast(), these pass arguments to contrast_calc(). For contrast_calc(), they are not used.
a, b	Lists containing conditions for all predictors in the model that will be contrasted to form the hypothesis H0: a = b. The gendata function will generate the necessary combinations and default values for unspecified predictors. See examples below.
cnames	A vector of character strings naming the contrasts when type = "individual". Usually cnames is not necessary as the function 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(40,50).
type	A character string. Set type="average" to average the individual contrasts (e.g., to obtain a "Type II" or "Type III" contrast).
weights	A numeric vector, used when type = "average", to obtain weighted contrasts.
conf.int	The confidence level for confidence intervals for the contrasts.
fcType	A character string: "simple", "log" or "signed".
fcFunc	A function to transform the numerator and denominator of fold changes.
covType	A string matching the method for estimating the covariance matrix. The default value produces the typical estimate. See sandwich::vcovHC() for options.
env	An environment in which evaluate fit.

Details

These functions mirror rms::contrast.rms() but have fewer options.

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/ratio) is returned.

Value

a list of class contrast.Design containing the elements Contrast, SE, Z, var, df.residual Lower, Upper, Pvalue, X, cnames, and foldChange, 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).

See Also

rms::contrast.rms(), sandwich::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)
)

# ------------------------------------------------------------------------

anova_model <- lm(expression ~ diet * group, data = two_factor_crossed)
anova(anova_model)

library(ggplot2)
theme_set(theme_bw() + theme(legend.position = "top"))
ggplot(two_factor_crossed) +
    aes(x = diet, y = expression, col = group, shape = group) +
    geom_point() +
    geom_smooth(aes(group = group), method = lm, se = FALSE)

int_model <- lm(expression ~ diet * group, data = two_factor_crossed)
main_effects <- lm(expression ~ diet + group, data = two_factor_crossed)

# Interaction effect is probably real:
anova(main_effects, int_model)

# Test treatment in low fat diet:
veh_group <- list(diet = "low fat", group = "vehicle")
trt_group <- list(diet = "low fat", group = "treatment")
contrast(int_model, veh_group, trt_group)

# ------------------------------------------------------------------------

car_mod <- lm(mpg ~ am + wt, data = mtcars)
print(summary(car_mod), digits = 5)

mean_wt <- mean(mtcars$wt)

manual_trans <- list(am = 0, wt = mean_wt)
auto_trans <- list(am = 1, wt = mean_wt)
print(contrast(car_mod, manual_trans, auto_trans), digits = 5)
```

`print.contrast` *Print a Contrast Object*

Description

Print a Contrast Object

Usage

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

Arguments

x	Result of `contrast()`.
X	A logical: set `TRUE` to print design matrix used in computing the contrasts (or the average contrast).
fun	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.
...	Not used.

`two_factor_crossed` *Complete Two-Factor Experiment*

Description

Complete Two-Factor Experiment

Details

A gene expression experiment was run to assess the effect of a compound under two different diets: high fat and low fat. The main comparisons of interest are the difference between the treated and untreated groups within a diet. The interaction effect was a secondary hypothesis. For illustration, we only include the expression value of one of the genes. The study design was a full two-way factorial with `n = 24` samples.

Value

`two_factor_crossed`
: A data frame

Examples

```
two_factor_crossed
```

`two_factor_incompl` *Incomplete Two-Factor Experiment with Repeated Measurments*

Description

Incomplete Two-Factor Experiment with Repeated Measurments

Details

In a gene expression experiment, stem cells were differentiated using a set of factors (such as media types, cell spreads etc.). These factors were collapsed into a single cell environment configurations variable. The cell lines were assays over three days. Two of the configurations were only run on the first day and the other two were assays at baseline.

To get the materials, three donors provided materials. These donors provided (almost) equal replication across the two experimental factors (day and configuration).

One of the goals of this experiment was to assess pre-specified differences in the configuration at each time point. For example, the differences between configurations `A` and `B` at day one is of interest. Also, the differences between configurations `C` and `D` at each time points were important.

Since there are missing cells in the design, it is not a complete two-way factorial. One way to analyze this experiment is to further collapse the time and configuration data into a single variable and then specify each comparison using this factor.

Value

`two_factor_incompl`
: A data frame

Examples

```
two_factor_incompl
```

Index

∗ **datasets**
    two_factor_crossed, 5
    two_factor_incompl, 6
∗ **models**
    contrast.lm, 2
∗ **regression**
    contrast.lm, 2

contrast.geese *(*contrast.lm*)*, 2
contrast.gls *(*contrast.lm*)*, 2
contrast.lm, 2
contrast.lme *(*contrast.lm*)*, 2
contrast_calc *(*contrast.lm*)*, 2

print.contrast, 4

rms::contrast.rms(), *3*

sandwich::vcovHC(), *3*

two_factor_crossed, 5
two_factor_incompl, 6