

Package ‘codingMatrices’

May 5, 2017

Type Package

Title Alternative Factor Coding Matrices for Linear Model Formulae

Version 0.3.1

Author Bill Venables

Maintainer Bill Venables <Bill.Venables@gmail.com>

Description A collection of coding functions as alternatives to the standard functions in the stats package, which have names starting with 'contr.'. Their main advantage is that they provide a consistent method for defining marginal effects in factorial models. In a simple one-way ANOVA model the intercept term is always the simple average of the class means.

Depends stats

Imports Matrix, methods, fractional, utils

Suggests knitr, MASS, dplyr, car, ggplot2

License GPL (>= 2)

LazyData TRUE

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2017-05-05 03:12:23 UTC

R topics documented:

Codings	2
mean_contrasts	4
Index	5

Description

These functions provide an alternative to the coding functions supplied in the `stats` package, namely `contr.treatment`, `contr.sum`, `contr.helmert` and `contr.poly`.

Usage

```
code_control(n, contrasts = TRUE, sparse = FALSE)
code_control_last(n, contrasts = TRUE, sparse = FALSE)
code_diff(n, contrasts = TRUE, sparse = FALSE)
code_diff_forward(n, contrasts = TRUE, sparse = FALSE)
code_helmert(n, contrasts = TRUE, sparse = FALSE)
code_helmert_forward(n, contrasts = TRUE, sparse = FALSE)
code_deviation(n, contrasts = TRUE, sparse = FALSE)
code_deviation_first(n, contrasts = TRUE, sparse = FALSE)
code_poly(n, contrasts = TRUE, sparse = FALSE)
contr.diff(n, contrasts = TRUE, sparse = FALSE)
```

Arguments

<code>n</code>	Either a positive integer giving the number of levels or the levels attribute of a factor, supplying both the number of levels via its length and labels potentially to be used in the <code>dimnames</code> of the result.
<code>contrasts</code>	Logical: Do you want the $n \times (n - 1)$ coding matrix (TRUE) or an $n \times n$ full-rank matrix, (as is sometimes needed by the fitting functions) (FALSE)?
<code>sparse</code>	Logical: Do you want the result to be a sparse matrix object, as generated the the <code>Matrix</code> package?

Details

All functions with names of the form `code_xxxx` return coding matrices which, in a simple model, make the intercept term the simple ("unweighted") average of the class means. This can be important in some non-standard ANOVA tables. The function `contr.diff` is an exception, and is offered as a natural companion to `stats::contr.treatment`, with which it is closely aligned.

- code_control** Similar to `contr.treatment`, with contrasts comparing the class means (the "treatments") with the first class mean (the "control").
- code_control_last** Similar to `code_control`, but using the final class mean as the "control". Cf. `contr.SAS`
- code_diff** The contrasts are the successive differences of the treatment means, $\mu_{i+i} - \mu_i$. This coding function has no counterpart in the stats package. It is suggested as an alternative to the default coding, `contr.poly`, for ordered factors. It offers a visual check of monotonicity of the class means with the ordered levels of the factor. Unlike `stats::contr.poly` there is no assumption that the factor levels are in some sense "equally spaced".
- code_diff_forward** Very similar to `code_diff`, but using forward differences: $\mu_i - \mu_{i+1}$
- code_helmert** Similar to `contr.helmert`, but with a small scaling change to make the regression coefficients (i.e. the contrasts) more easily interpretable. The contrasts now compare each class mean, starting from the second, with the average of all class means coming prior to it in the factor levels order.
- code_helmert_forward** Similar to `code_helmert`, but comparing each class mean, up to the second last, with the average of all class means coming after it in the factor levels order.
- code_deviation** Similar to `contr.sum`, which is described as having the "effects" summing to zero. A more precise description might be to say that the contrasts are the deviations of each class mean from the average of them, i.e. $\mu_i - \bar{\mu}$. To avoid redundancy, the last deviation is omitted.
- code_deviation_first** Very similar to `code_deviation`, but omitting the first deviation to avoid redundancy rather than the last.
- code_poly** Similar in effect to `contr.poly` but for levels fewer than 15 using an unnormalized basis for the orthogonal polynomials with integer entries. (Orthogonal polynomials were originally given in this form as tables.) The only advantage over `stats::contr.poly` is one of display. Use `stats::contr.poly` in preference other than for teaching purposes.
- contr.diff** Very similar in effect to `code_diff`, yielding the same differences as the contrasts, but like `stats::contr.treatment` using the first class mean as the intercept coefficient rather than the simple average of the class means, as with `code_diff`. Some would regard this as making it unsuitable for use in some non-standard ANOVA tables.

Value

A coding matrix, as requested by fitting functions using linear model formulae with factor predictors.

See Also

The MASS function `contr.sdif` which is an early version of `code_deviation` (by the same author).

Examples

```
(M <- code_control(5))
mean_contrasts(M)
(M <- stats::contr.treatment(5))
mean_contrasts(M) ## same contrasts; different averaging vector.
mean_contrasts(stats::contr.helmert(6)) ## Interpretation obscure
mean_contrasts(code_helmert(6))      ## each mean with the average preceding
mean_contrasts(code_helmert_forward(6)) ## each mean with the average succeeding
```

mean_contrasts	<i>Display contrasts</i>
----------------	--------------------------

Description

A function to display the averaging vector and class mean contrasts implied by a given coding matrix

Usage

```
mean_contrasts(M)
```

Arguments

M Any $n \times (n - 1)$ coding matrix.

Value

The full contrast matrix, `solve(cbind(1, M))`, suitably annotated and presented in vulgar fractional form, for clarity.

Examples

```
mean_contrasts(code_helmert_forward(5))  
mean_contrasts(code_diff_forward(letters[1:7]))
```

Index

`code_control` (Codings), 2
`code_control_last` (Codings), 2
`code_deviation` (Codings), 2
`code_deviation_first` (Codings), 2
`code_diff` (Codings), 2
`code_diff_forward` (Codings), 2
`code_helmert` (Codings), 2
`code_helmert_forward` (Codings), 2
`code_poly` (Codings), 2
Codings, 2
`contr.diff` (Codings), 2
`contr.helmert`, 3
`contr.poly`, 3
`contr.SAS`, 3
`contr.sdif`, 3
`contr.sum`, 3
`contr.treatment`, 3

`mean_contrasts`, 4