Package ‘hypr’

February 28, 2020

Type Package
Title Hypothesis Matrix Translation
URL https://maxrabe.com/hypr
BugReports https://github.com/mmrabe/hypr/issues
Version 0.1.6
License GPL-3
Depends R (>= 3.5.0)
Imports MASS, pracma, methods
Suggests knitr, rmarkdown
VignetteBuilder knitr
Encoding UTF-8
LazyData true
RoxygenNote 7.0.2
Collate 'equations.R' 'hypr.R'
NeedsCompilation no
Author Maximilian M. Rabe [aut, cre],
Shravan Vasishth [aut],
Sven Hohenstein [aut],
Reinhold Kliegl [aut],
Daniel J. Schad [aut]
Maintainer Maximilian M. Rabe <maximilian.rabe@uni-potsdam.de>
Repository CRAN
Date/Publication 2020-02-28 09:40:02 UTC
R topics documented:

\[+,\text{hypr,}\text{hypr-method}\]

\[\text{cmat}\]

\[\text{conversions}\]

\[\text{formula<-}\]

\[\text{ginv2}\]

\[\text{hmat}\]

\[\text{hypr}\]

\[\text{hypr-class}\]

Index

\[+,\text{hypr,}\text{hypr-method}\]

### Combining hypr objects by addition or interaction

#### Description

You can combine one or more hypr objects, i.e. combine their hypothesis to a single hypr object, by adding them with the \(+\) or \(\times\) operators.

#### Usage

```
## S4 method for signature 'hypr,hypr'
e1 + e2
## S4 method for signature 'hypr,hypr'
e1 * e2
## S4 method for signature 'hypr,hypr'
e1 & e2
## S4 method for signature 'hypr,hypr'
e1 / e2
```

#### Arguments

- \(e1\), \(e2\)  
  hypr objects to concatenate

#### Value

The combined hypr object

#### Functions

- \(+,\text{hypr,}\text{hypr-method}\): Interaction of \(e1\) and \(e2\)
- \(\&,\text{hypr,}\text{hypr-method}\): Interaction and main contrasts of \(e1\) and \(e2\)
- \(\div,\text{hypr,}\text{hypr-method}\): Nesting levels of \(e2\) within \(e1\)
Examples

(h1 <- hypr(a~i, b~i)) # a hypr object of two treatments
(h2 <- hypr(i~0)) # an intercept-only hypr object
hc <- h1 + h2
hc
interaction <- h1 & h2
interaction_and_main <- h1 * h2

cmat Retrieve or set contrast matrix

Description

Use these functions to retrieve or set a hypr object's contrast matrix. If used for updating, the hypothesis matrix and equations are derived automatically.

Usage

cmat(x, add_intercept = FALSE, remove_intercept = FALSE, as_fractions = TRUE)
cmat(x, add_intercept = FALSE, remove_intercept = FALSE) <- value
contr.hypothesis(
  ..., 
  add_intercept = FALSE,
  remove_intercept = NULL,
  as_fractions = FALSE
)

Arguments

x A hypr object
add_intercept Add additional intercept column to contrast matrix
remove_intercept If TRUE, tries to find an intercept column (all codes equal) and removes it from the matrix. If NULL, does the same but does not throw an exception if no intercept is found. FALSE explicitly disables this functionality. A numeric argument explicitly identifies the index of the column to be removed.
as_fractions Should the returned matrix be formatted as fractions (using MASS::as.fractions())?
value contrast matrix
... A list of hypothesis equations for which to retrieve a contrast matrix
Details

Basic specification of contrasts in R is accomplished with basic R functions `stats::contrasts()` and `stats::C()` (Chambers & Hastie, 1992). Other relevant packages for this topic are `multcomp` (Bretz et al., 2010), `contrast` (Kuhn et al., 2016), and, including also various vignettes, `emmeans` (Lenth, 2019).

Value

A matrix of contrast codes with contrasts as columns and levels as rows.

Functions

- `cmat<-`: Set contrast matrix
- `contr.hypothesis`: Retrieve contrast matrix to override factor contrasts

References


See Also

`hypr`

Examples

```r
h <- hypr(mu1~0, mu2~mu1)
cmat(h) # retrieve the contrast matrix
contr.hypothesis(h) # by default without intercept (removes first column)
contr.hypothesis(mu1~0, mu2~mu1)
```
Shorthand versions for simple hypothesis translation

Description

These functions can be used to translate between null hypothesis equations, hypothesis matrices, and contrast matrices without defining a hypr object. Note that some of these functions do generate a hypr object internally but they never return one.

Usage

```r
eqs2hmat(
  eqs,
  levels = NULL,
  order_levels = missing(levels),
  as_fractions = TRUE
)

eqs2cmat(eqs, as_fractions = TRUE)

hmat2cmat(hmat, as_fractions = TRUE)

cmat2hmat(cmat, as_fractions = TRUE)

hmat2eqs(hmat, as_fractions = TRUE)

cmat2eqs(cmat, as_fractions = TRUE)
```

Arguments

- **eqs**: A list of equations
- **levels**: (optional) A character vector of variables to be expected (if not provided, automatically generated from all terms occurring in the equations list)
- **order_levels**: (optional) Whether to alphabetically order appearance of levels (rows in transposed hypothesis matrix or contrast matrix). Default is `TRUE` if `levels` were not explicitly provided.
- **as_fractions**: (optional) Whether to output matrix using fractions formatting (via `MASS::as.fractions`). Defaults to `TRUE`.
- **hmat**: Hypothesis matrix
- **cmat**: Contrast matrix

Value

A list of equations (`hmat2eqs` and `cmat2eqs`), a contrast matrix (`hmat2cmat`, `eqs2cmat`), or a hypothesis matrix (`cmat2hmat`, `eqs2hmat`).
Functions

- eqs2hmat: Convert null hypothesis equations to hypothesis matrix
- eqs2cmat: Convert null hypothesis equations to contrast matrix
- hmat2cmat: Convert hypothesis matrix to contrast matrix
- cmat2hmat: Convert contrast matrix to hypothesis matrix
- hmat2eqs: Convert hypothesis matrix to null hypothesis equations
- cmat2eqs: Convert contrast matrix to null hypothesis equations

Examples

# The following examples are based on a 2-level treatment contrast (i.e., baseline and treatment).
# hypotheses <- list(baseline = mu1~0, treatment = mu2~mu1)
# hypothesis_matrix <- matrix(
#   c(c(1, -1), c(0, 1)), ncol = 2, dimnames = list(c("baseline", "treatment"), c("mu1", "mu2")))
# contrast_matrix <- matrix(
#   c(c(1, 1), c(0, 1)), ncol = 2, dimnames = list(c("mu1", "mu2"), c("baseline", "treatment")))

# Convert a list of null hypothesis equations to ...
# ... a hypothesis matrix:
eqs2hmat(hypotheses)
# ... a contrast matrix:
eqs2cmat(hypotheses)

# Convert a hypothesis matrix to...
# ... a list of null hypothesis equations:
hmat2eqs(hypothesis_matrix)
# ... a contrast matrix:
hmat2cmat(hypothesis_matrix)

# Convert a contrast matrix to...
# ... a list of null hypothesis equations:
cmat2eqs(contrast_matrix)
# ... a hypothesis matrix:
cmat2hmat(contrast_matrix)

# Are all functions returning the expected results?
stopifnot(all.equal(eqs2hmat(hypotheses, as_fractions = FALSE), hypothesis_matrix))
stopifnot(all.equal(eqs2cmat(hypotheses, as_fractions = FALSE), contrast_matrix))
stopifnot(all.equal(hmat2cmat(hypothesis_matrix, as_fractions = FALSE), contrast_matrix))
stopifnot(all.equal(cmat2hmat(contrast_matrix, as_fractions = FALSE), hypothesis_matrix))

Description
This is a generic function for setting an S4 object’s formulas.

Usage
formula(x, ...) <- value

Arguments
x The object to manipulate
... Additional arguments passed on to the method
value The new formula

ginv2
Enhanced generalized inverse function

Description
This function is a wrapper for MASS::ginv and calculates the generalized inverse of x.

Usage
ginv2(x, as_fractions = TRUE)

Arguments
x The original matrix
as_fractions Whether to format the matrix as fractions (MASS package)

Details
In addition to MASS::ginv, this function rounds values, formats the matrix as fractions and copies dimension names from the original matrix.

Value
Generalized inverse of x

See Also
ginv
Examples

```r
h <- hypr(mu1~0, mu2~mu1)
hmat(h)

ginv2(hmat(h))
cmat(h)

# cmat is effectively the generalized inverse of hmat
stopifnot(all.equal(ginv2(hmat(h)), cmat(h)))
```

---

### hmat

**Retrieve and set hypothesis matrix**

#### Description

Use these functions to retrieve or set a hypr object’s hypothesis matrix. If used for updating, the contrast matrix and equations are derived automatically.

#### Usage

- `hmat(x, as_fractions = TRUE)`
- `thmat(x, as_fractions = TRUE)`
- `hmat(x) <- value`
- `thmat(x) <- value`

#### Arguments

- `x`: A hypr object
- `as_fractions`: Whether to format matrix as fractions (via `MASS::as.fractions`)
- `value`: Hypothesis matrix

#### Value

Hypothesis matrix of `x`

#### Functions

- `thmat`: Retrieve transposed hypothesis matrix
- `hmat<`: Set hypothesis matrix
- `thmat<`: Set transposed hypothesis matrix
Examples

```r
h <- hypr(mu1~0, mu2~mu1)

# To retrieve the hypothesis matrix of `h`:
hm(h)

# To retrieve the transposed hypothesis matrix of `h`:
thm(h)

# Setting the hypothesis matrix of `h`:
h <- matrix(c(1,-1,0,1), ncol=2, dimnames=list(NULL, c("mu1","mu2")))

h

h2 <- hypr() # an empty hypr object

thm(h2) <- matrix(c(1,0,-1,1), ncol=2, dimnames=list(c("mu1","mu2"), NULL))

h2

# `h' and `h2' should be identical:
stopifnot(all.equal(hm(h), hm(h2)))
stopifnot(all.equal(cm(h), cm(h2)))
```
Details
You may call the function without any arguments. In that case, an empty hypr object is returned. This is useful if you want to derive equations from a known hypothesis matrix or contrast matrix.

Basic specification of contrasts in R is accomplished with basic R functions `stats::contrasts()` and `stats::C()` (Chambers & Hastie, 1992). Other relevant packages for this topic are multcomp (Bretz et al., 2010), contrast (Kuhn et al., 2016), and, including also various vignettes, emmeans (Lenth, 2019).

Value
A hypr object

References

Frank Bretz, Torsten Hothorn and Peter Westfall (2010), Multiple Comparisons Using R, CRC Press, Boca Raton.


See Also
`contrasts` and `C` for basic specification of contrasts in R, S4 class `hypr`, `cmat`, `contr.hypothesis` for retrieval of contrast matrices from `hypr` objects

Examples

# Create an empty hypr object (no hypotheses):
hypr()

# Treatment contrast:
hypr(mu1 ~ 0, mu2 ~ mu1, mu3 ~ mu1, mu4 ~ mu1)

# Identical version:
hypr(-mu1, -mu2 + mu1, -mu3 + mu1, -mu4 + mu1)

cContr.hypothesis(h)

# Generate a dataset
set.seed(123)
M <- c(mu1 = 10, mu2 = 20, mu3 = 10, mu4 = 40) # condition means
N <- 5 # number of observations per condition
SD <- 10 # residual SD
```r
simdat <- do.call(rbind, lapply(names(M), function(x) {
  data.frame(X = x, DV = as.numeric(MASS::mvrnorm(N, unname(M[x]), SD^2, empirical = TRUE)))
}))
simdat$X <- factor(simdat$X, levels=levels(h))
simdat

# Check agreement of hypothesis levels and factor levels
stopifnot(levels(h) == levels(simdat$X))

# Linear regression
contrasts(simdat$X) <- contr.hypothesis(h)
round(coef(summary(lm(DV ~ X, data=simdat))),3)
```

---

**S4 class “hypr” and its methods**

### Description

A `hypr` object contains equations, a hypothesis matrix and a contrast matrix, all of which are related to each other. See below for methods.

### Usage

#### # S4 method for signature 'hypr'
```
show(object)
```

#### # S4 method for signature 'hypr'
```
levels(x)
```

#### # S4 method for signature 'hypr'
```
names(x)
```

#### # S4 replacement method for signature 'hypr'
```
names(x) <- value
```

#### # S4 replacement method for signature 'hypr'
```
levels(x) <- value
```

#### # S4 method for signature 'hypr'
```
formula(x, ...)
```

#### # S4 replacement method for signature 'hypr'
```
formula(x, ...) <- value
```
Arguments

- object, x: a hypr object
- value: New value (list of equations for formula, character vector for levels and names)
- ... (ignored)

Details

To generate a hypr object, use the `hypr` function.

Value

A character vector of level names
A character vector of contrast names
A list of null hypothesis equations

Methods (by generic)

- `show`: Show summary of hypr object, including contrast equations, the (transposed) hypothesis matrix and the derived contrast matrix.
- `levels`: Retrieve the levels (variable names) used in a hypr object
- `names`: Retrieve the contrast names used in a hypr object
- `names<-`: Set the contrast names used in a hypr object
- `levels<-`: Set the levels used in a hypr object
- `formula`: Retrieve a hypr object’s null hypothesis equations.
- `formula<-`: Modify a hypr object’s null hypothesis equations

Slots

- `eqs`: List of null hypotheses
- `hmat`: Hypothesis matrix
- `cmat`: Contrast matrix

See Also

`hypr`, `cmat`, `hmat`

Examples

```r
# Equations and matrices in a hypr object are always congruent
# Therefore creating a hypr object h and then copying ...
h <- hypr(mu1~0, mu2~mu1)

# ... its equations, ...
h2 <- hypr()
formula(h2) <- formula(h)
```
# ... its hypothesis matrix, ...
h3 <- hypr()
hmat(h3) <- hmat(h)

# ... or its contrast matrix ...
h4 <- hypr()
cmat(h4) <- cmat(h)

# ... over to another hypr object is the same as copying the object:
h5 <- h

# check that hypr objects are equal by comparing hmat() and cmat()
stopifnot(all.equal(hmat(h), hmat(h2)))
stopifnot(all.equal(cmat(h), cmat(h2)))
stopifnot(all.equal(hmat(h), hmat(h3)))
stopifnot(all.equal(cmat(h), cmat(h3)))
stopifnot(all.equal(hmat(h), hmat(h4)))
stopifnot(all.equal(cmat(h), cmat(h4)))
stopifnot(all.equal(hmat(h), hmat(h5)))
stopifnot(all.equal(cmat(h), cmat(h5)))

h <- hypr(mu1~0, mu2~mu1)
formula(h)

h2 <- hypr()
formula(h2) <- formula(h)
h2
formula(h2)

# After updating, matrices should be equal
stopifnot(all.equal(hmat(h), hmat(h2)))
stopifnot(all.equal(cmat(h), cmat(h2)))
Index

*, hypr, hypr-method
  (+, hypr, hypr-method), 2
+, hypr, hypr-method, 2
/, hypr, hypr-method
  (+, hypr, hypr-method), 2
&, hypr, hypr-method
  (+, hypr, hypr-method), 2

C, 10
  cmat, 3, 10, 12
  cmat2eqs (conversions), 5
  cmat2hmat (conversions), 5
  cmat<-(cmat), 3
  contr.hypothesis, 10
  contr.hypothesis (cmat), 3
  contrasts, 10
  conversions, 5

  eqs2cmat (conversions), 5
  eqs2hmat (conversions), 5

  formula, hypr-method (hypr-class), 11
  formula<-, 6
  formula<-, hypr-method (hypr-class), 11

  ginv, 7
  ginv2, 7

  hmat, 8, 12
  hmat2cmat (conversions), 5
  hmat2eqs (conversions), 5
  hmat<-(hmat), 8
  hypr, 4, 9, 10, 12
  hypr-class, 11

  levels, hypr-method (hypr-class), 11
  levels<-, hypr-method (hypr-class), 11

  MASS::as.fractions, 5, 8
  MASS::as.fractions(), 3
  names, hypr-method (hypr-class), 11
  names<-, hypr-method (hypr-class), 11
  show, hypr-method (hypr-class), 11
  stats::C(), 4, 10
  stats::contrasts(), 4, 10
  thmat (hmat), 8
  thmat<-(hmat), 8