

Package ‘heplots’

January 2, 2012

Type Package

Title Visualizing Tests in Multivariate Linear Models

Version 0.9-10

Date 2011-09-12

Author John Fox, Michael Friendly, and Georges Monette

Maintainer Michael Friendly <friendly@yorku.ca>

Depends R (>= 2.10), car, graphics, stats

Suggests rgl, candisc, effects, reshape, gplots, nlme, lattice

LazyLoad yes

LazyData yes

Description Provides HE plot functions for visualizing hypothesis tests in multivariate linear models. They represents sums-of-squares-and-products matrices for linear hypotheses and for error using ellipses (in two dimensions) and ellipsoids (in three dimensions).

License GPL (>= 2)

Repository CRAN

Repository/R-Forge/Project heplots

Repository/R-Forge/Revision 114

Date/Publication 2011-10-17 05:54:56

R topics documented:

heplots-package	2
Adopted	5
arrow3d	6
Bees	7
cross3d	9
ellipse3d.axes	10
etasq	11
FootHead	13
gsorth	14
Headache	16
heplot	18
heplot1d	23
heplot3d	26
Hernior	30
mark.H0	31
MockJury	32
pairs.mlm	34
Plastic	36
RatWeight	37
ReactTime	38
Rohwer	39
RootStock	41
SocGrades	42
statList	43
termMeans	45
trans.colors	46
VocabGrowth	47
WeightLoss	48
Index	51

heplots-package

Visualizing Tests in Multivariate Linear Models

Description

The heplots package provides functions for visualizing hypothesis tests in multivariate linear models (MANOVA, multivariate multiple regression, MANCOVA, and repeated measures designs). They represent sums-of-squares-and-products matrices for linear hypotheses and for error using ellipses (in two dimensions), ellipsoids (in three dimensions), or by line segments in one dimension. See Fox, Friendly and Monette (2007) for a brief introduction.

The package also provides a collection of data sets illustrating a variety of multivariate linear models of the types listed above, together with graphical displays.

Several tutorial vignettes are also included.

Details

Package: heplots
 Type: Package
 Version: 0.9-10
 Date: 2011-09-12
 License: GPL (>= 2), GPL version 2 or newer

The graphical functions contained here all display multivariate model effects in variable (data) space, for one or more response variables (or contrasts among response variables in repeated measures designs).

`heplot` constructs two-dimensional HE plots for model terms and linear hypotheses for pairs of response variables in multivariate linear models.

`heplot3d` constructs analogous 3D plots for triples of response variables.

`pairs.mlm` constructs a “matrix” of pairwise HE plots.

`heplot1d` constructs 1-dimensional analogs of HE plots for model terms and linear hypotheses for single response variables.

For repeated measure designs, between-subject effects and within-subject effects must be plotted separately, because the error terms (E matrices) differ. For terms involving within-subject effects, these functions carry out a linear transformation of the matrix \mathbf{Y} of responses to a matrix $\mathbf{Y}\mathbf{M}$, where \mathbf{M} is the model matrix for a term in the intra-subject design and produce plots of the H and E matrices in this transformed space. The vignette `repeated` describes these graphical methods for repeated measures designs.

The related `car-package` package calculates Type II and Type III tests of multivariate linear hypotheses using the `Anova` and `linearHypothesis` functions.

The `candisc-package` package provides functions for visualizing effects for model terms in a low-dimensional canonical space that shows the largest hypothesis relative to error variation.

Author(s)

John Fox, Michael Friendly, and Georges Monette

Maintainer: Michael Friendly, <friendly@yorku.ca>, <http://datavis.ca>

References

Friendly, M. (2006). Data Ellipses, HE Plots and Reduced-Rank Displays for Multivariate Linear Models: SAS Software and Examples. *Journal of Statistical Software*, 17(6), 1-42. <http://www.jstatsoft.org/v17/i06/>

Friendly, M. (2007). HE plots for Multivariate General Linear Models. *Journal of Computational and Graphical Statistics*, 16(2) 421-444. <http://datavis.ca/papers/jcgs-heplots.pdf>

Fox, J., Friendly, M. & Monette, G. (2007). Visual hypothesis tests in multivariate linear models: The heplots package for R. *DSC 2007: Directions in Statistical Computing*. <http://socserv.socsci.mcmaster.ca/jfox/heplots-dsc-paper.pdf>

Friendly, Michael (2010). HE Plots for Repeated Measures Designs. *Journal of Statistical Software*, 37(4), 1-40. URL <http://www.jstatsoft.org/v37/i04/>.

See Also

[Anova](#), [linearHypothesis](#) for Anova.mlm computations and tests
[candisc-package](#) for reduced-rank views in canonical space
[manova](#) for a different approach to testing effects in MANOVA designs

Adopted

Adopted Children

Description

Data are a subset from an observational, longitudinal, study on adopted children. Is child's intelligence related to intelligence of the biological mother and the intelligence of the adoptive mother?

The child's intelligence was measured at age 2, 4, 8, and 13 for this sample. How does intelligence change over time, and how are these changes related to intelligence of the birth and adoptive mother?

Usage

Adopted

Format

A data frame with 62 observations on the following 6 variables.

AMED adoptive mother's years of education (proxy for her IQ)

BMIQ biological mother's score on IQ test

Age2IQ IQ of child at age 2

Age4IQ IQ of child at age 4

Age8IQ IQ of child at age 8

Age13IQ IQ of child at age 13

Source

Ramsey, F.L. and Schafer, D.W. (2002). *The Statistical Sleuth: A Course in Methods of Data Analysis (2nd ed)*, Duxbury.

This data set is identical to [ex1605](#) in the Sleuth2 package.

References

Friendly, Michael (2010). HE Plots for Repeated Measures Designs. *Journal of Statistical Software*, 37(4), 1-40. URL <http://www.jstatsoft.org/v37/i04/>.

Skodak, M. and Skeels, H.M. (1949). A Final Follow-up Study of One Hundred Adopted Children, *Journal of Genetic Psychology* **75**: 85-125.

See Also[ex1605](#)**Examples**

```

# Treat as multivariate regression problem
Adopted.mod <- lm(cbind(Age2IQ, Age4IQ, Age8IQ, Age13IQ) ~ AMED + BMIQ, data=Adopted)
Adopted.mod

# test overall multivariate regression
linearHypothesis(Adopted.mod, c("AMED", "BMIQ"))

# show separate linear regressions
op <- par(mfcol=c(2,4), mar=c(4,4,2,2)+.1)
for (i in 3:6) {
  dataEllipse(as.matrix(Adopted[,c(1,i)]), col="black", levels=0.68, ylim=c(70,140))
  abline(lm(Adopted[,i] ~ Adopted[,1]), col="red", lwd=2)
  dataEllipse(as.matrix(Adopted[,c(2,i)]), col="black", levels=0.68, ylim=c(70,140))
  abline(lm(Adopted[,i] ~ Adopted[,2]), col="red", lwd=2)
  abline(a=0,b=1, lty=1, col="blue")
}
par(op)

# between-S (MMReg) plots
heplot(Adopted.mod, hypotheses=list("Reg"=c("AMED", "BMIQ")),
main="IQ scores of adopted children: MMReg")
pairs(Adopted.mod, hypotheses=list("Reg"=c("AMED", "BMIQ")))

heplot3d(Adopted.mod, hypotheses=list("Reg"=c("AMED", "BMIQ")),
col = c("pink", "blue", "black", "gray"))

# Treat IQ at different ages as a repeated measure factor
# within-S models & plots
Age <- data.frame(Age=ordered(c(2,4,8,13)))
Anova(Adopted.mod, idata=Age, idesign=~Age, test="Roy")

# within-S plots
heplot(Adopted.mod, idata=Age, idesign=~Age, iterm="Age",
hypotheses=list("Reg"=c("AMED", "BMIQ")))
)

```

arrow3d

Draw a 3D Arrow in an RGL Scene

Description

Draws a 3D arrow in an rgl scene with barbs at the arrow head

Usage

```
arrow3d(p0 = c(0, 0, 0), p1 = c(1, 1, 1), barblen, s = 0.05, theta = pi/6, n = 3, ...)
```

Arguments

p0	Initial point (tail of arrow)
p1	Ending point (head of arrow)
barblen	Length of each barb, in data units
s	length of barb as fraction of line length (unless barblen is specified)
theta	opening angle of barbs
n	number of barbs
...	args passed to lines3d for line styling, e.g., color, lwd, etc. See material3d .

Value

Returns (invisibly): integer ID of the line added to the scene

Author(s)

Barry Rowlingson, posted to R-help, 1/10/2010

See Also

[lines3d](#), [segments3d](#),

Examples

```
arrow3d(c(0,0,0), c(2,2,2), barblen=.2, lwd=3, col="black")
arrow3d(c(0,0,0), c(-2,2,2), barblen=.2, lwd=3, col="red")
```

Bees

Captive and maltreated bees

Description

Pabalan, Davey and Packe (2000) studied the effects of captivity and maltreatment on reproductive capabilities of queen and worker bees in a complex factorial design.

Bees were placed in a small tube and either held captive (CAP) or shaken periodically (MAL) for one of 5, 7.5, 10, 12.5 or 15 minutes, after which they were sacrificed and two measures: ovarian development (Iz) and ovarian reabsorption (Iy), were taken. A single control group was measured with no such treatment, i.e., at time 0; there are n=10 per group.

The design is thus nearly a three-way factorial, with factors caste (Queen, Worker), treat (CAP, MAL) and time, except that there are only 11 combinations of Treatment and Time; we call these trtime below.

Usage

```
data(Bees)
```

Format

A data frame with 246 observations on the following 6 variables.

`caste` a factor with levels Queen Worker

`treat` a factor with levels "" CAP MAL

`time` an ordered factor: time of treatment

`Iz` an index of ovarian development

`Iy` an index of ovarian reabsorption

`trtime` a factor with levels 0 CAP05 CAP07 CAP10 CAP12 CAP15 MAL05 MAL07 MAL10 MAL12 MAL15

Details

Models for the three-way factorial design, using the formula `cbind(Iz, Iy) ~ caste*treat*time` ignore the control condition at `time==0`, where `treat==NA`.

To handle the additional control group at `time==0`, while separating the effects of Treatment and Time, 10 contrasts can be defined for the `trtime` factor in the model `cbind(Iz, Iy) ~ caste*trtime`. See `demo(bees.contrasts)` for details.

In the `heplot` examples below, the default `size="evidence"` displays are too crowded to interpret, because some effects are so highly significant. The alternative effect-size scaling, `size="effect"`, makes the relations clearer.

Source

Pabalan, N., Davey, K. G. & Packe, L. (2000). Escalation of Aggressive Interactions During Staged Encounters in *Halictus ligatus* Say (Hymenoptera: Halictidae), with a Comparison of Circle Tube Behaviors with Other Halictine Species *Journal of Insect Behavior*, **13**, 627-650.

References

Friendly, M. (2006). Data Ellipses, HE Plots and Reduced-Rank Displays for Multivariate Linear Models: SAS Software and Examples *Journal of Statistical Software*, **17**, 1-42.

Examples

```
data(Bees)

# 3-way factorial, ignoring 0 group
bees.mod <- lm(cbind(Iz, Iy) ~ caste*treat*time, data=Bees)
Anova(bees.mod)

op<-palette(c(palette()[1:4], "brown", "magenta", "olivedrab", "darkgray"))
heplot(bees.mod, xlab="Iz: Ovarian development", ylab="Iz: Ovarian reabsorption",
main="Bees: ~caste*treat*time")
heplot(bees.mod, xlab="Iz: Ovarian development", ylab="Iz: Ovarian reabsorption",
```

```

      main="Bees: ~caste*treat*time", size="effect")

# two-way design, using trtime
bees.mod1 <- lm(cbind(Iz,Iy) ~ caste*trtime, data=Bees)
Anova(bees.mod1)

heplot(bees.mod1, xlab="Iz: Ovarian development", ylab="Iz: Ovarian reabsorption",
main="Bees: ~caste*trtime")
heplot(bees.mod1, xlab="Iz: Ovarian development", ylab="Iz: Ovarian reabsorption",
      main="Bees: ~caste*trtime", size="effect")
palette(op)

# effect plots for separate responses
if(require(effects)) {
bees.lm1 <-lm(Iy ~ treat*caste*time, data=Bees)
bees.lm2 <-lm(Iz ~ treat*caste*time, data=Bees)

bees.eff1 <- allEffects(bees.lm1)
plot(bees.eff1,multiline=TRUE,ask=FALSE)

bees.eff2 <- allEffects(bees.lm2)
plot(bees.eff2,multiline=TRUE,ask=FALSE)
}

```

cross3d

Draw a 3D cross in an rgl scene

Description

Draws a 3D cross or axis vectors in an rgl scene.

Usage

```
cross3d(centre = rep(0, 3), scale = rep(1, 3), ...)
```

Arguments

centre	A scalar or vector of length 3, giving the centre of the 3D cross
scale	A scalar or vector of length 3, giving the lengths of the arms of the 3D cross
...	Other arguments, passed on to segments3d

Value

Used for its side-effect, but returns (invisibly) a 6 by 3 matrix containing the end-points of three axes, in pairs.

Author(s)

Michael Friendly

See Also[segments3d](#)

<code>ellipse3d.axes</code>	<i>Draw axes of a 3D ellipsoid</i>
-----------------------------	------------------------------------

Description

A function to draw the major axes of a 3D ellipsoid from a correlation, covariance or sums of squares and cross products matrix.

Usage

```
ellipse3d.axes(x, centre = c(0, 0, 0), scale = c(1, 1, 1),
              level = 0.95, t = sqrt(qchisq(level, 3)), which = 1:3,
              labels = TRUE, label.ends=c(2,4,6), ...)
```

Arguments

<code>x</code>	A square positive definite matrix at least 3x3 in size. It will be treated as the correlation or covariance of a multivariate normal distribution.
<code>centre</code>	The center of the ellipse
<code>scale</code>	If <code>x</code> is a correlation matrix, then the standard deviations of each parameter can be given in the scale parameter. This defaults to <code>c(1, 1, 1)</code> , so no rescaling will be done.
<code>level</code>	The confidence level of a simultaneous confidence region. The default is 0.95, for a 95% region. This is used to control the size of the ellipsoid.
<code>t</code>	The size of the ellipsoid may also be controlled by specifying the value of a t-statistic on its boundary.
<code>which</code>	This parameter selects which variables from the object will be plotted. The default is the first 3.
<code>labels</code>	Either a logical value, a character string, or a character vector of length 3. If TRUE, the default, the axes are labeled PC1, PC2, PC3. If a single character string, the digits 1, 2, 3 are pasted on the end.
<code>label.ends</code>	A vector of length 3 indicating which ends of the axes should be labeled, corresponding to a selection of rows of the 6 x 3 matrix of axes end points. Default: <code>c(2,4,6)</code> .
<code>...</code>	Other arguments passed to <code>segments3d</code> and <code>text3d</code> .

Value

Returns a 6 x 3 matrix containing the end points of the three axis lines in pairs by rows.

Author(s)

Michael Friendly

See Also[segments3d](#), [text3d](#), [ellipse3d](#)**Examples**

```
data(iris)
iris3 <- iris[,1:3]
cov <- cov(iris3)
mu <- colMeans(iris3)
col <- c("blue", "green", "red")[iris$Species]

library(rgl)
plot3d(iris3, type="s", size=0.4, col=col, cex=2, box=FALSE, aspect="iso")
plot3d( ellipse3d(cov, centre=mu, level=0.68), col="gray", alpha=0.2, add = TRUE)

axes <- ellipse3d.axes(cov, centre=mu, level=0.68, color="gray", lwd=2)
```

etasq

Measures of Partial Association (Eta-squared) for Linear Models

Description

Calculates partial eta-squared for linear models or multivariate analogs of eta-squared (or R^2), indicating the partial association for each term in a multivariate linear model. There is a different analog for each of the four standard multivariate test statistics: Pillai's trace, Hotelling-Lawley trace, Wilks' Lambda and Roy's maximum root test.

Usage

```
etasq(x, ...)
```

```
## S3 method for class 'lm'
etasq(x, anova = FALSE, partial = TRUE, ...)
```

```
## S3 method for class 'mlm'
etasq(x, ...)
```

```
## S3 method for class 'Anova.mlm'
etasq(x, anova = FALSE, ...)
```

Arguments

x	A lm, mlm or Anova.mlm object
anova	A logical, indicating whether the result should also contain the test statistics produced by Anova().
partial	A logical, indicating whether to calculate partial or classical η^2 .
...	Other arguments passed down to Anova .

Details

For univariate linear models, classical $\eta^2 = SSH / SST$ and partial $\eta^2 = SSH / (SSH + SSE)$. These are identical in one-way designs.

Partial eta-squared describes the proportion of total variation attributable to a given factor, partialling out (excluding) other factors from the total nonerror variation. These are commonly used as measures of effect size or measures of (non-linear) strength of association in ANOVA models.

All multivariate tests are based on the $s = \min(p, df_h)$ latent roots of HE^{-1} . The analogous multivariate partial η^2 measures are calculated as:

Pillai's trace (V) $\eta^2 = V/s$

Hotelling-Lawley trace (T) $\eta^2 = T/(T + s)$

Wilks' Lambda (L) $\eta^2 = L^{1/s}$

Roy's maximum root (R) $\eta^2 = R/(R + 1)$

Value

When anova=FALSE, a one-column data frame containing the eta-squared values for each term in the model.

When anova=TRUE, a 5-column (lm) or 7-column (mlm) data frame containing the eta-squared values and the test statistics produced by `print.Anova()` for each term in the model.

Author(s)

Michael Friendly

References

Muller, K. E. and Peterson, B. L. (1984). Practical methods for computing power in testing the Multivariate General Linear Hypothesis *Computational Statistics and Data Analysis*, 2, 143-158.

Muller, K. E. and LaVange, L. M. and Ramey, S. L. and Ramey, C. T. (1992). Power Calculations for General Linear Multivariate Models Including Repeated Measures Applications. *Journal of the American Statistical Association*, 87, 1209-1226.

See Also

[Anova](#)

Examples

```
data(Soils) # from car package
soils.mod <- lm(cbind(pH,N,Dens,P,Ca,Mg,K,Na,Conduc) ~ Block + Contour*Depth, data=Soils)
#Anova(soils.mod)
etasq(Anova(soils.mod))
etasq(soils.mod) # same
etasq(Anova(soils.mod), anova=TRUE)

etasq(soils.mod, test="Wilks")
etasq(soils.mod, test="Hotelling")
```

FootHead

Head measurements of football players

Description

Data collected as part of a preliminary study examining the relation between football helmet design and neck injuries. There are 30 subjects in each of three groups: High school football players, college players and non-football players.

Usage

```
data(FootHead)
```

Format

A data frame with 90 observations on the following 7 variables.

group a factor with levels High school College Non-football

width a numeric vector: head width at widest dimension

circum a numeric vector: head circumference

front.back a numeric vector: front to back distance at eye level

eye.top a numeric vector: eye to top of head

ear.top a numeric vector: ear to top of head

jaw a numeric vector: jaw width

Source

Rencher, A. C. (1995), *Methods of Multivariate Analysis*, New York: Wiley, Table 8.3.

Examples

```

data(FootHead)
str(FootHead)

# use Helmert contrasts for group
contrasts(FootHead$group) <- contr.helmert
contrasts(FootHead$group)

foot.mod <- lm(cbind(width, circum, front.back, eye.top, ear.top, jaw)~group, data=FootHead)
Manova(foot.mod)

# show the HE plot for the first two variables
heplot(foot.mod, main="HE plot for width and circumference", fill=TRUE,
col=c("red", "blue"))

# show it with tests of Helmert contrasts
heplot(foot.mod, hypotheses=list("group.1"="group1", "group.2"="group2"),
col=c("red", "blue", "green3", "green3" ),
main="HE plot with orthogonal Helmert contrasts")

# show all pairwise HE plots
pairs(foot.mod)

# ... with tests of Helmert contrasts
pairs(foot.mod, hypotheses=list("group.1"="group1", "group.2"="group2"),
col=c("red", "blue", "green3", "green3"), hyp.labels=FALSE)

# see that the hypothesis for groups really is 2D
heplot3d(foot.mod, variables=c(1,2,6),
hypotheses=list("group.1"="group1", "group.2"="group2"),
col=c("red", "blue", "green3", "green3" ), wire=FALSE)

```

gsorth

Orthogonalize successive columns of a data frame or matrix

Description

gsorth uses sequential, orthogonal projections, as in the Gram-Schmidt method, to transform a matrix or numeric columns of a data frame into an uncorrelated set, possibly retaining the same column means and standard deviations as the original.

Usage

```
gsorth(y, order, recenter = TRUE, rescale = TRUE, adjnames = TRUE)
```

Arguments

<code>y</code>	A numeric data frame or matrix
<code>order</code>	An integer vector specifying the order of and/or a subset of the columns of <code>y</code> to be orthogonalized. If missing, <code>order=1:p</code> where <code>p=ncol(y)</code> .
<code>recenter</code>	If TRUE, the result has same column means as original; else means = 0 for cols 2:p.
<code>rescale</code>	If TRUE, the result has same column standard deviations as original; else sd = residual variance for cols 2:p
<code>adjnames</code>	If TRUE, the column names of the result are adjusted to the form Y1, Y2.1, Y3.12, by adding the suffixes '.1', '.12', etc. to the original column names.

Details

The method is equivalent to setting each of columns 2:p to the residuals from a linear regression of that column on all prior columns, i.e.,

```
z[,j] <- resid( lm( z[,j] ~ as.matrix(z[,1:(j-1)]), data=z ) )
```

However, for accuracy and speed the transformation is carried out using the QR decomposition.

Value

Returns a matrix or data frame with uncorrelated columns. Row and column names are copied to the result.

Author(s)

Michael Friendly

See Also

[qr](#),

Examples

```
GSiris <- gsorth(iris[,1:4])
GSiris <- gsorth(iris, order=1:4) # same, using order
str(GSiris)
zapsmall(cor(GSiris))
colMeans(GSiris)
# sd(GSiris) -- sd(<matrix>) now deprecated
apply(GSiris, 2, sd)

# orthogonalize Y side
GSiris <- data.frame(gsorth(iris[,1:4]), Species=iris$Species)
iris.mod1 <- lm(as.matrix(GSiris[,1:4]) ~ Species, data=GSiris)
Anova(iris.mod1)

# orthogonalize X side
rohwer.mod <- lm(cbind(SAT, PPVT, Raven) ~ n + s + ns + na + ss, data=Rohwer)
```

```

Anova(rohwer.mod)

# type I tests for Rohwer data
Rohwer.orth <- cbind(Rohwer[,1:5], gsorth(Rohwer[, c("n", "s", "ns", "na", "ss")], adjnames=FALSE))

rohwer.mod1 <- lm(cbind(SAT, PPVT, Raven) ~ n + s + ns + na + ss, data=Rohwer.orth)
Anova(rohwer.mod1)
# compare with anova()
anova(rohwer.mod1)

```

 Headache

Treatment of Headache Sufferers for Sensitivity to Noise

Description

A study was conducted investigating the effectiveness of different kinds of psychological treatment on the sensitivity of headache sufferers to noise, described in Hand and Taylor (1987), Study E.

In a pre-post design, 98 patients were first assessed for the volume of noise which they found uncomfortable (U) and definitely uncomfortable (DU). They were then given relaxation training, where they listened to the noise at the DU level and given instruction breathing techniques and the use of visual imagery to distract them from discomfort. One of four treatments was then applied, and all patients were reassessed for the noise volume they considered uncomfortable (U) and definitely uncomfortable (DU).

Usage

```
data(Headache)
```

Format

A data frame with 98 observations on the following 6 variables.

`type` Type of headache, a factor with levels Migrane Tension

`treatment` Treatment group, a factor with levels T1 T2 T3 Control. See Details

`u1` Noise level rated as Uncomfortable, initial measure

`du1` Noise level rated as Definitely Uncomfortable, initial measure

`u2` Noise level rated as Uncomfortable, final measure

`du2` Noise level rated as Definitely Uncomfortable, final measure

Details

The treatments are described as follows:

T1 Listened again to the tone at their initial DU level, for the same amount of time they were able to tolerate it before.

T2 Same as T1, with one additional minute exposure

T3 Same as T2, but were explicitly instructed to use the relaxation techniques

Control These subject experienced no further exposure to the noise tone until the final sensitivity measures were taken

Hand and Taylor described several substantive hypotheses related to the differences among treatments. In the Headache data frame, these have been included as contrasts (Headache\$treatment)

Source

D. J. Hand and C. C. Taylor (1987). *Multivariate analysis of variance and repeated measures: a practical approach for behavioural scientists* London: Chapman and Hall. ISBN: 0412258005. Table E.1.

Examples

```
data(Headache)
str(Headache)

# basic MLM, specifying between-S effects
headache.mod <- lm(cbind(u1, du1, u2, du2) ~ type * treatment, data=Headache)

#####
## between-S tests
#####
Anova(headache.mod, test="Roy")

# test each contrast separately
print(linearHypothesis(headache.mod, hypothesis="treatment1", test="Roy"), SSP=FALSE)
print(linearHypothesis(headache.mod, hypothesis="treatment2", test="Roy"), SSP=FALSE)
print(linearHypothesis(headache.mod, hypothesis="treatment3", test="Roy"), SSP=FALSE)

heplot(headache.mod, variables=c(1,3),
hypotheses=paste("treatment", 1:3, sep=""),
hyp.labels=c("extra.exp", "no.inst", "explicit.inst"),
xlab="u1: Initial sensitivity", ylab="u2: Final sensitivity",
main="Headache data: Unpleasant levels")
abline(0, 1, lty=5, col="gray")

heplot(headache.mod, variables=c(2,4),
hypotheses=paste("treatment", 1:3, sep=""),
xlab="du1: Initial sensitivity", ylab="du2: Final sensitivity",
main="Headache data: Definitely Unpleasant levels")
abline(0, 1, lty=5, col="gray")
```

```

pairs(headache.mod)

#####
# between-S and within-S tests
#####
idata = expand.grid(level=factor(c("U", "DU")), phase=factor(1:2))
Anova(headache.mod, idata=idata, idesign=~level*phase)

```

heplot

Two-Dimensional HE Plots

Description

This function plots ellipses representing the hypothesis and error sums-of-squares-and-products matrices for terms and linear hypotheses in a multivariate linear model. These include MANOVA models (all explanatory variables are factors), multivariate regression (all quantitative predictors), MANCOVA models, homogeneity of regression, as well as repeated measures designs treated from a multivariate perspective.

Usage

```

heplot(mod, ...)

## S3 method for class 'mlm'
heplot(mod, terms, hypotheses, term.labels = TRUE,
  hyp.labels = TRUE, err.label="Error", variables = 1:2, error.ellipse = !add,
  factor.means = !add, grand.mean = !add, remove.intercept = TRUE,
  type = c("II", "III", "2", "3"), idata=NULL, idesign=NULL,
  icontrasts=c("contr.sum", "contr.poly"), imatrix=NULL, iterm=NULL, markH0=!is.null(ityerm),
  manova, size = c("evidence", "effect.size"),
  level = 0.68, alpha = 0.05, segments = 40,
  center.pch = "+", center.cex=2,
  col = palette()[-1],
  lty = 2:1, lwd = 1:2,
  fill=FALSE, fill.alpha=0.3,
  xlab, ylab, main = "", xlim, ylim, axes=TRUE, offset.axes,
  add = FALSE, verbose = FALSE, warn.rank = FALSE, ...)

```

Arguments

mod	a model object of class "mlm".
terms	a logical value or character vector of terms in the model for which to plot hypothesis matrices; if missing or TRUE, defaults to all terms; if FALSE, no terms are plotted.
hypotheses	optional list of linear hypotheses for which to plot hypothesis matrices; hypotheses are specified as for the linear.hypothesis function in the car package; the list elements can be named, in which case the names are used.

<code>term.labels</code>	logical value or character vector of names for the terms to be plotted. If TRUE (the default) the names of the terms are used; if FALSE, term labels are not plotted.
<code>hyp.labels</code>	logical value or character vector of names for the hypotheses to be plotted. If TRUE (the default) the names of components of the list of hypotheses are used; if FALSE, hypothesis labels are not plotted.
<code>err.label</code>	Label for the error ellipse
<code>variables</code>	indices or names of the two response variables to be plotted; defaults to 1:2.
<code>error.ellipse</code>	if TRUE, plot the error ellipse; defaults to TRUE, if the argument <code>add</code> is FALSE (see below).
<code>factor.means</code>	logical value or character vector of names of factors for which the means are to be plotted, or TRUE or FALSE; defaults to TRUE, if the argument <code>add</code> is FALSE (see below).
<code>grand.mean</code>	if TRUE, plot the centroid for all of the data; defaults to TRUE, if the argument <code>add</code> is FALSE (see below).
<code>remove.intercept</code>	if TRUE (the default), do not plot the ellipse for the intercept even if it is in the MANOVA table.
<code>type</code>	“type” of sum-of-squares-and-products matrices to compute; one of “II”, “III”, “2”, or “3”, where “II” is the default (and “2” is a synonym).
<code>idata</code>	an optional data frame giving a factor or factors defining the intra-subject model for multivariate repeated-measures data. See Details of Anova for an explanation of the intra-subject design and for further explanation of the other arguments relating to intra-subject factors.
<code>idesign</code>	a one-sided model formula using the “data” in <code>idata</code> and specifying the intra-subject design for repeated measure models.
<code>icontrasts</code>	names of contrast-generating functions to be applied by default to factors and ordered factors, respectively, in the within-subject “data”; the contrasts must produce an intra-subject model matrix in which different terms are orthogonal. The default is <code>c("contr.sum", "contr.poly")</code> .
<code>imatrix</code>	In lieu of <code>idata</code> and <code>idesign</code> , you can specify the intra-subject design matrix directly via <code>imatrix</code> , in the form of list of named elements. Each element gives the columns of the within-subject model matrix for an intra-subject term to be tested, and must have as many rows as there are responses; the columns of the within-subject model matrix for <i>different</i> terms must be mutually orthogonal. <i>This functionality requires car version 2.0 or later.</i>
<code>iterm</code>	For repeated measures designs, you must specify one intra-subject term (a character string) to select the SSPE (E) matrix used in the HE plot. Hypothesis terms plotted include the <code>iterm</code> effect as well as all interactions of <code>iterm</code> with terms.
<code>markH0</code>	A logical value (or else a list of arguments to <code>mark.H0</code>) used to draw cross-hairs and a point indicating the value of a point null hypothesis. The default is TRUE if <code>iterm</code> is non-NULL.
<code>manova</code>	optional <code>Anova.mlm</code> object for the model; if absent a MANOVA is computed. Specifying the argument can therefore save computation in repeated calls.

<code>size</code>	how to scale the hypothesis ellipse relative to the error ellipse; if "evidence", the default, the scaling is done so that a "significant" hypothesis ellipse at level α extends outside of the error ellipse; if "effect.size", the hypothesis ellipse is on the same scale as the error ellipse.
<code>level</code>	equivalent coverage of ellipse for normally-distributed errors, defaults to 0.68.
<code>alpha</code>	significance level for Roy's greatest-root test statistic; if <code>size="evidence"</code> , then the hypothesis ellipse is scaled so that it just touches the error ellipse at the specified alpha level; a larger hypothesis ellipse therefore indicates statistical significance; defaults to 0.05.
<code>segments</code>	number of line segments composing each ellipse; defaults to 40.
<code>center.pch</code>	character to use in plotting the centroid of the data; defaults to "+".
<code>center.cex</code>	size of character to use in plotting the centroid of the data; defaults to 2.
<code>col</code>	a colour or vector of colours to use in plotting ellipses; the first colour is used for the error ellipse; the remaining colours — recycled as necessary — are used for the hypothesis ellipses; a single colour can be given, in which case it is used for all ellipses. Defaults to the current colour palette, less its first element.
<code>lty</code>	vector of line types to use for plotting the ellipses; the first is used for the error ellipse, the rest — possibly recycled — for the hypothesis ellipses; a single line type can be given. Defaults to 2:1.
<code>lwd</code>	vector of line widths to use for plotting the ellipses; the first is used for the error ellipse, the rest — possibly recycled — for the hypothesis ellipses; a single line width can be given. Defaults to 1:2.
<code>fill</code>	A logical vector indicating whether each ellipse should be filled or not. The first value is used for the error ellipse, the rest — possibly recycled — for the hypothesis ellipses; a single fill value can be given. Defaults to FALSE for backward compatibility. See Details below.
<code>fill.alpha</code>	Alpha transparency for filled ellipses, a numeric scalar or vector of values within $[0, 1]$, where 0 means fully transparent and 1 means fully opaque. Defaults to 0.3.
<code>xlab</code>	x-axis label; defaults to name of the x variable.
<code>ylab</code>	y-axis label; defaults to name of the y variable.
<code>main</code>	main plot label; defaults to "".
<code>xlim</code>	x-axis limits; if absent, will be computed from the data.
<code>ylim</code>	y-axis limits; if absent, will be computed from the data.
<code>axes</code>	Whether to draw the x, y axes; defaults to TRUE
<code>offset.axes</code>	proportion to extend the axes in each direction if computed from the data; optional.
<code>add</code>	if TRUE, add to the current plot; the default is FALSE. If TRUE, the error ellipse is not plotted.
<code>verbose</code>	if TRUE, print the MANOVA table and details of hypothesis tests; the default is FALSE.
<code>warn.rank</code>	if TRUE, do not suppress warnings about the rank of the hypothesis matrix when the ellipse collapses to a line; the default is FALSE.
<code>...</code>	arguments to pass down to <code>plot</code> , <code>text</code> , and <code>points</code> .

Details

The `heplot` function plots a representation of the covariance ellipses for hypothesized model terms and linear hypotheses (H) and the corresponding error (E) matrices for two response variables in a multivariate linear model (mlm).

The plot helps to visualize the nature and dimensionality response variation on the two variables jointly in relation to error variation that is summarized in the various multivariate test statistics (Wilks' Lambda, Pillai trace, Hotelling-Lawley trace, Roy maximum root). Roy's maximum root test has a particularly simple visual interpretation, exploited in the `size="evidence"` version of the plot. See the description of argument `alpha`.

For a 1 df hypothesis term (a quantitative regressor, a single contrast or parameter test), the H matrix has rank 1 (one non-zero latent root of HE^{-1}) and the H "ellipse" collapses to a degenerate line.

Typically, you fit a mlm with `mymlm <- lm(cbind(y1, y2, y3, ...) ~ modelterms)`, and plot some or all of the `modelterms` with `heplot(mymlm, ...)`. Arbitrary linear hypotheses related to the terms in the model (e.g., contrasts of an effect) can be included in the plot using the `hypotheses` argument. See [linear.hypothesis](#) for details.

For repeated measure designs, where the response variables correspond to one or more variates observed under a within-subject design, between-subject effects and within-subject effects must be plotted separately, because the error terms (E matrices) differ. When you specify an intra-subject term (`iterm`), the analysis and HE plots amount to analysis of the matrix **Y** of responses post-multiplied by a matrix **M** determined by the intra-subject design for that term. See Friendly (2010) or the vignette("repeated") in this package for an extended discussion and examples.

The related [candisc](#) package provides functions for visualizing a multivariate linear model in a low-dimensional view via a generalized canonical discriminant analyses. [heplot.candisc](#) and [heplot3d.candisc](#) provide a low-rank 2D (or 3D) view of the effects for a given term in the space of maximum discrimination.

When an element of `fill` is TRUE, the ellipse outline is drawn using the corresponding color in `col`, and the interior is filled with a transparent version of this color specified in `fill.alpha`. To produce filled (non-degenerate) ellipses without the bounding outline, use a value of `lty=0` in the corresponding position.

Value

The function invisibly returns an object of class "heplot", with coordinates for the various hypothesis ellipses and the error ellipse, and the limits of the horizontal and vertical axes. These may be useful for adding additional annotations to the plot, using standard plotting functions. (No methods for manipulating these objects are currently available.)

The components are:

H	a list containing the coordinates of each ellipse for the hypothesis terms
E	a matrix containing the coordinates for the error ellipse
center	x,y coordinates of the centroid
xlim	x-axis limits
ylim	y-axis limits
radius	the radius for the unit circles used to generate the ellipses

References

- Friendly, M. (2006). Data Ellipses, HE Plots and Reduced-Rank Displays for Multivariate Linear Models: SAS Software and Examples *Journal of Statistical Software*, **17**(6), 1–42. <http://www.jstatsoft.org/v17/i06/>
- Friendly, M. (2007). HE plots for Multivariate General Linear Models. *Journal of Computational and Graphical Statistics*, **16**(2) 421–444. <http://datavis.ca/papers/jcgs-heplots.pdf>
- Friendly, Michael (2010). HE Plots for Repeated Measures Designs. *Journal of Statistical Software*, **37**(4), 1-40. URL <http://www.jstatsoft.org/v37/i04/>.

See Also

[Anova](#), [linearHypothesis](#) for details on testing mlms.

[heplot1d](#), [heplot3d](#), [pairs.mlm](#), [mark.H0](#) for other HE plot functions. [trans.colors](#) for calculation of transparent colors.

[candisc](#), [heplot.candisc](#) for reduced-rank views of mlms in canonical space.

Examples

```
## iris data
contrasts(iris$Species)<-matrix(c(0,-1,1, 2, -1, -1), 3,2)
contrasts(iris$Species)

iris.mod <- lm(cbind(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) ~
Species, data=iris)

hyp <- list("V:V"="Species1", "S:VV"="Species2")
heplot(iris.mod, hypotheses=hyp)
# compare with effect-size scaling
heplot(iris.mod, hypotheses=hyp, size="effect", add=TRUE)

# try filled ellipses
heplot(iris.mod, hypotheses=hyp, fill=TRUE, col=c("red", "blue"))
heplot(iris.mod, hypotheses=hyp, fill=TRUE, col=c("red", "blue"), lty=c(0,0,1,1))

hep <-heplot(iris.mod, variables=c(1,3), hypotheses=hyp)
str(hep)

# all pairs
pairs(iris.mod, hypotheses=hyp, hyp.labels=FALSE)

## Pottery data, from car package
data(Pottery)
pottery.mod <- lm(cbind(Al, Fe, Mg, Ca, Na) ~ Site, data=Pottery)
heplot(pottery.mod)
heplot(pottery.mod, terms=FALSE, add=TRUE, col="blue",
hypotheses=list(c("SiteCaldicot = 0", "SiteIsleThorns=0")),
hyp.labels="Sites Caldicot and Isle Thorns")

## Rohwer data, multivariate multiple regression/ANCOVA
```

```

#-- ANCOVA, assuming equal slopes
rohwer.mod <- lm(cbind(SAT, PPVT, Raven) ~ SES + n + s + ns + na + ss, data=Rohwer)
Anova(rohwer.mod)
col <- c("red", "black", "blue", "cyan", "magenta", "brown", "gray")
heplot(rohwer.mod, col=col)
# Add ellipse to test all 5 regressors
heplot(rohwer.mod, hypotheses=list("Regr" = c("n", "s", "ns", "na", "ss")), col=col, fill=TRUE)
# View all pairs
pairs(rohwer.mod, hypotheses=list("Regr" = c("n", "s", "ns", "na", "ss")))
# or 3D plot
col <- c("pink", "black", "blue", "cyan", "magenta", "brown", "gray")
heplot3d(rohwer.mod, hypotheses=list("Regr" = c("n", "s", "ns", "na", "ss")), col=col)

```

heplot1d

One-Dimensional HE Plots

Description

This function plots a 1-dimensional representation of the hypothesis (H) and error (E) sums-of-squares-and-products matrices for terms and linear hypotheses in a multivariate linear model.

In particular, for a given response, the 1-D representations of H and E matrices correspond to line segments. The E “ellipse” is shown as a filled rectangle whose width equals the mean squared error for that response. The H “ellipse” for each model term is shown as a line segment whose length represents either the size of the effect or the evidence for that effect.

This version is an initial sketch. Details of the implementation are subject to change.

Usage

```
heplot1d(mod, ...)
```

```

## S3 method for class 'mlm'
heplot1d(mod, terms, hypotheses, term.labels = TRUE, hyp.labels = TRUE,
variables = 1, error.ellipse = !add, factor.means = !add, grand.mean = !add,
remove.intercept = TRUE, type = c("II", "III", "2", "3"),
idata=NULL, idesign=NULL, icontrasts=c("contr.sum", "contr.poly"), imatrix=NULL, iterm=NULL, manova,
size = c("evidence", "effect.size"), level = 0.68, alpha = 0.05,
center.pch = "|", col = palette()[-1], lty = 2:1, lwd = 1:2, xlab, main = "",
xlim, axes = TRUE, offset.axes, add = FALSE, verbose = FALSE, ...)

```

Arguments

mod	a model object of class "mlm".
terms	a logical value or character vector of terms in the model for which to plot hypothesis matrices; if missing or TRUE, defaults to all terms; if FALSE, no terms are plotted.
hypotheses	optional list of linear hypotheses for which to plot hypothesis matrices; hypotheses are specified as for the linearHypothesis function in the car package; the list elements can be named, in which case the names are used.

<code>term.labels</code>	logical value or character vector of names for the terms to be plotted. If TRUE (the default) the names of the terms are used; if FALSE, term labels are not plotted.
<code>hyp.labels</code>	logical value or character vector of names for the hypotheses to be plotted. If TRUE (the default) the names of components of the list of hypotheses are used; if FALSE, hypothesis labels are not plotted.
<code>variables</code>	indices or names of the two response variables to be plotted; defaults to 1:2.
<code>error.ellipse</code>	if TRUE, plot the error ellipse; defaults to TRUE, if the argument <code>add</code> is FALSE (see below).
<code>factor.means</code>	logical value or character vector of names of factors for which the means are to be plotted, or TRUE or FALSE; defaults to TRUE, if the argument <code>add</code> is FALSE (see below).
<code>grand.mean</code>	if TRUE, plot the centroid for all of the data; defaults to TRUE, if the argument <code>add</code> is FALSE (see below).
<code>remove.intercept</code>	if TRUE (the default), do not plot the ellipse for the intercept even if it is in the MANOVA table.
<code>type</code>	“type” of sum-of-squares-and-products matrices to compute; one of “II”, “III”, “2”, or “3”, where “II” is the default (and “2” is a synonym).
<code>idata</code>	an optional data frame giving a factor or factors defining the intra-subject model for multivariate repeated-measures data. See Details of Anova for an explanation of the intra-subject design and for further explanation of the other arguments relating to intra-subject factors.
<code>idesign</code>	a one-sided model formula using the “data” in <code>idata</code> and specifying the intra-subject design for repeated measure models.
<code>icontrasts</code>	names of contrast-generating functions to be applied by default to factors and ordered factors, respectively, in the within-subject “data”; the contrasts must produce an intra-subject model matrix in which different terms are orthogonal. The default is <code>c("contr.sum", "contr.poly")</code> .
<code>imatrix</code>	In lieu of <code>idata</code> and <code>idesign</code> , you can specify the intra-subject design matrix directly via <code>imatrix</code> , in the form of list of named elements. Each element gives the columns of the within-subject model matrix for an intra-subject term to be tested, and must have as many rows as there are responses; the columns of the within-subject model matrix for <i>different</i> terms must be mutually orthogonal. <i>This functionality requires car version 2.0 or later.</i>
<code>iterm</code>	For repeated measures designs, you must specify one intra-subject term (a character string) to select the SSPE (E) matrix used in the HE plot. Hypothesis terms plotted include the <code>iterm</code> effect as well as all interactions of <code>iterm</code> with terms.
<code>manova</code>	optional <code>Anova.mlm</code> object for the model; if absent a MANOVA is computed. Specifying the argument can therefore save computation in repeated calls.
<code>size</code>	how to scale the hypothesis ellipse relative to the error ellipse; if “evidence”, the default, the scaling is done so that a “significant” hypothesis ellipse extends outside of the error ellipse; if “effect.size”, the hypothesis ellipse is on the same scale as the error ellipse.
<code>level</code>	equivalent coverage of ellipse for normally-distributed errors, defaults to 0.68.

<code>alpha</code>	significance level for Roy's greatest-root test statistic; if <code>size="evidence"</code> , then the hypothesis ellipse is scaled so that it just touches the error ellipse at the specified alpha level; a larger hypothesis ellipse therefore indicates statistical significance; defaults to 0.05.
<code>center.pch</code>	character to use in plotting the centroid of the data; defaults to <code>" "</code> .
<code>col</code>	a colour or vector of colours to use in plotting ellipses; the first colour is used for the error ellipse; the remaining colours — recycled as necessary — are used for the hypothesis ellipses; a single colour can be given, in which case it is used for all ellipses. Defaults to the current colour palette, less its first element.
<code>lty</code>	vector of line types to use for plotting the ellipses; the first is used for the error ellipse, the rest — possibly recycled — for the hypothesis ellipses; a single line type can be given. Defaults to 2:1.
<code>lwd</code>	vector of line widths to use for plotting the ellipses; the first is used for the error ellipse, the rest — possibly recycled — for the hypothesis ellipses; a single line width can be given. Defaults to 1:2.
<code>xlab</code>	x-axis label; defaults to name of the x variable.
<code>main</code>	main plot label; defaults to <code>""</code> .
<code>xlim</code>	x-axis limits; if absent, will be computed from the data.
<code>axes</code>	Whether to draw the x, y axes; defaults to TRUE
<code>offset.axes</code>	proportion to extend the axes in each direction if computed from the data; optional.
<code>add</code>	if TRUE, add to the current plot; the default is FALSE. If TRUE, the error ellipse is not plotted.
<code>verbose</code>	if TRUE, print the MANOVA table and details of hypothesis tests; the default is FALSE.
<code>...</code>	arguments to pass down to <code>plot</code> , <code>text</code> , and <code>points</code> .

Value

The function invisibly returns an object of class `"heplot1d"`, with coordinates for the various hypothesis ellipses and the error ellipse, and the limits of the horizontal and vertical axes. (No methods for manipulating these objects are currently available.)

The components are:

<code>H</code>	ranges for the hypothesis terms
<code>E</code>	range for E
<code>xlim</code>	x-axis limits

Author(s)

Michael Friendly

See Also

[Anova](#), [linearHypothesis](#) for hypothesis tests in `mlms`
[heplot](#), [heplot3d](#), [pairs.mlm](#) for other HE plot methods

Examples

```
## Plastics data
plastic.mod <- lm(cbind(tear, gloss, opacity) ~ rate*additive, data=Plastic)
heplot1d(plastic.mod, col=c("pink","blue"))
heplot1d(plastic.mod, col=c("pink","blue"),variables=2)
heplot1d(plastic.mod, col=c("pink","blue"),variables=3)

## Bees data
bees.mod <- lm(cbind(Iz,Iy) ~ caste*treat*time, data=Bees)
heplot1d(bees.mod)
heplot1d(bees.mod, variables=2)
```

heplot3d

Three-Dimensional HE Plots

Description

This function plots ellipsoids in 3D representing the hypothesis and error sums-of-squares-and-products matrices for terms and linear hypotheses in a multivariate linear model.

Usage

```
heplot3d(mod, ...)
```

```
## S3 method for class 'mlm'
heplot3d(mod, terms, hypotheses, term.labels = TRUE,
  hyp.labels = TRUE, err.label="Error", variables = 1:3, error.ellipsoid = !add,
  factor.means = !add, grand.mean = !add, remove.intercept = TRUE,
  type = c("II", "III", "2", "3"), idata=NULL, idesign=NULL,
  icontrasts=c("contr.sum", "contr.poly"), imatrix=NULL, iterm=NULL,
  manova, size = c("evidence", "effect.size"),
  level = 0.68, alpha = 0.05, segments = 40, col = palette()[-1], lwd=c(1, 4),
  shade=TRUE, shade.alpha=0.1, wire=c(TRUE,FALSE),
  bg.col = c("white", "black"), fogtype = c("none", "exp2", "linear", "exp"),
  fov = 30, offset = 0.01, xlab, ylab, zlab, xlim, ylim, zlim, add = FALSE, verbose = FALSE,
  warn.rank = FALSE, ...)
```

Arguments

<code>mod</code>	a model object of class "mlm".
<code>terms</code>	a logical value or character vector of terms in the model for which to plot hypothesis matrices; if missing or TRUE, defaults to all terms; if FALSE, no terms are plotted.
<code>hypotheses</code>	optional list of linear hypotheses for which to plot hypothesis matrices; hypotheses are specified as for the linearHypothesis function in the car package; the list elements can be named, in which case the names are used.

<code>term.labels</code>	logical value or character vector of names for the terms to be plotted. If TRUE (the default) the names of the terms are used; if FALSE, term labels are not plotted.
<code>hyp.labels</code>	logical value or character vector of names for the hypotheses to be plotted. If TRUE (the default) the names of components of the list of hypotheses are used; if FALSE, hypothesis labels are not plotted.
<code>err.label</code>	Label for the error ellipse
<code>variables</code>	indices or names of the three response variables to be plotted; defaults to 1:3.
<code>error.ellipsoid</code>	if TRUE, plot the error ellipsoid; defaults to TRUE, if the argument <code>add</code> is FALSE (see below).
<code>factor.means</code>	logical value or character vector of names of factors for which the means are to be plotted, or TRUE or FALSE; defaults to TRUE, if the argument <code>add</code> is FALSE (see below).
<code>grand.mean</code>	if TRUE, plot the centroid for all of the data; defaults to TRUE, if the argument <code>add</code> is FALSE (see below).
<code>remove.intercept</code>	if TRUE (the default), do not plot the ellipsoid for the intercept even if it is in the MANOVA table.
<code>type</code>	“type” of sum-of-squares-and-products matrices to compute; one of “II”, “III”, “2”, or “3”, where “II” is the default (and “2” is a synonym).
<code>idata</code>	an optional data frame giving a factor or factors defining the intra-subject model for multivariate repeated-measures data. See Details of Anova for an explanation of the intra-subject design and for further explanation of the other arguments relating to intra-subject factors.
<code>idesign</code>	a one-sided model formula using the “data” in <code>idata</code> and specifying the intra-subject design for repeated measure models.
<code>icontrasts</code>	names of contrast-generating functions to be applied by default to factors and ordered factors, respectively, in the within-subject “data”; the contrasts must produce an intra-subject model matrix in which different terms are orthogonal. The default is <code>c("contr.sum", "contr.poly")</code> .
<code>imatrix</code>	In lieu of <code>idata</code> and <code>idesign</code> , you can specify the intra-subject design matrix directly via <code>imatrix</code> , in the form of list of named elements. Each element gives the columns of the within-subject model matrix for an intra-subject term to be tested, and must have as many rows as there are responses; the columns of the within-subject model matrix for <i>different</i> terms must be mutually orthogonal. <i>This functionality requires car version 2.0 or later.</i>
<code>iterm</code>	For repeated measures designs, you must specify one intra-subject term (a character string) to select the SSPE (E) matrix used in the HE plot. Hypothesis terms plotted include the <code>iterm</code> effect as well as all interactions of <code>iterm</code> with terms.
<code>manova</code>	optional <code>Anova.mlm</code> object for the model; if absent a MANOVA is computed. Specifying the argument can therefore save computation in repeated calls.
<code>size</code>	how to scale the hypothesis ellipsoid relative to the error ellipsoid; if “evidence”, the default, the scaling is done so that a “significant” hypothesis ellipsoid extends outside of the error ellipsoid; if “effect.size”, the hypothesis ellipsoid is on the same scale as the error ellipsoid.

level	equivalent coverage of ellipsoid for normally-distributed errors, defaults to 0.68.
alpha	significance level for Roy's greatest-root test statistic; if size="evidence", then the hypothesis ellipsoid is scaled so that it just touches the error ellipsoid at the specified alpha level; a larger hypothesis ellipsoid therefore indicates statistical significance; defaults to 0.05.
segments	number of segments composing each ellipsoid; defaults to 40.
col	a colour or vector of colours to use in plotting ellipsoids; the first colour is used for the error ellipsoid; the remaining colours — recycled as necessary — are used for the hypothesis ellipsoids; a single colour can be given, in which case it is used for all ellipsoids. Defaults to the current colour palette, less its first element.
lwd	a two-element vector giving the line width for drawing ellipsoids (including those that degenerate to an ellipse) and for drawing ellipsoids that degenerate to a line segment. The default is c(1, 4).
shade	a logical scalar or vector, indicating whether the ellipsoids should be rendered with shade3d . Works like col, except that FALSE is used for any 1 df degenerate ellipsoid.
shade.alpha	a numeric value in the range [0,1], or a vector of such values, giving the alpha transparency for ellipsoids rendered with shade=TRUE.
wire	a logical scalar or vector, indicating whether the ellipsoids should be rendered with wire3d . Works like col, except that TRUE is used for any 1 df degenerate ellipsoid.
bg.col	background colour, "white" or "black", defaulting to "white".
fogtype	type of "fog" to use for depth-cueing; the default is "none". See bg .
fov	field of view angle; controls perspective. See viewpoint .
offset	proportion of axes to off set labels; defaults to 0.01.
xlab	x-axis label; defaults to name of the x variable.
ylab	y-axis label; defaults to name of the y variable.
zlab	z-axis label; defaults to name of the z variable.
xlim	x-axis limits; if absent, will be computed from the data.
ylim	y-axis limits; if absent, will be computed from the data.
zlim	z-axis limits; if absent, will be computed from the data.
add	if TRUE, add to the current plot; the default is FALSE. If TRUE, the error ellipsoid is neither plotted nor returned in the output object.
verbose	if TRUE, print the MANOVA table and details of hypothesis tests; the default is FALSE.
warn.rank	if TRUE, do not suppress warnings about the rank of the hypothesis matrix when the ellipsoid collapses to an ellipse or line; the default is FALSE.
...	arguments passed from generic.

Details

When the H matrix for a term has rank < 3 , the ellipsoid collapses to an ellipse (rank(H)=2) or a line (rank(H)=1).

Rotating the plot can be particularly revealing, showing views in which H variation is particularly large or small in relation to E variation. See [play3d](#) and [movie3d](#) for details on creating animations.

The arguments `xlim`, `ylim`, and `zlim` can be used to expand the bounding box of the axes, but cannot decrease it.

Value

`heplot3d` invisibly returns a list containing the bounding boxes of the error (E) ellipsoid and for each term or linear hypothesis specified in the call. Each of these is a 2 x 3 matrix with rownames "min" and "max" and colnames corresponding to the variables plotted. An additional component, `center`, contains the coordinates of the centroid in the plot.

The function also leaves an object named `.frame` in the global environment, containing the `rgl` object IDs for the axes, axis labels, and bounding box; these are deleted and the axes, etc. redrawn if the plot is added to.

References

Friendly, M. (2006). Data Ellipses, HE Plots and Reduced-Rank Displays for Multivariate Linear Models: SAS Software and Examples *Journal of Statistical Software*, 17(6), 1-42. <http://www.jstatsoft.org/v17/i06/>

Friendly, M. (2007). HE plots for Multivariate General Linear Models. *Journal of Computational and Graphical Statistics*, 16(2) 421-444. <http://datavis.ca/papers/jcgs-heplots.pdf>

See Also

[Anova](#), [linearHypothesis](#), for details on MANOVA tests and linear hypotheses

[heplot](#), [pairs.mlm](#), for other plotting methods for `mlm` objects

[rgl-package](#), for details about 3D plots with `rgl`

[heplot3d.candisc](#) for 3D HE plots in canonical space.

Examples

```
# Soils data, from car package
soils.mod <- lm(cbind(pH,N,Dens,P,Ca,Mg,K,Na,Conduc) ~ Block + Contour*Depth, data=Soils)
Anova(soils.mod)
```

```
heplot(soils.mod, variables=c("Ca", "Mg"))
pairs(soils.mod, terms="Depth", variables=c("pH", "N", "P", "Ca", "Mg"))
```

```
heplot3d(soils.mod, variables=c("Mg", "Ca", "Na"), wire=FALSE)
```

```
# Plastic data
plastic.mod <- lm(cbind(tear, gloss, opacity) ~ rate*additive, data=Plastic)
heplot3d(plastic.mod, col=c("red", "blue", "brown", "green3"), wire=FALSE)
```

Hernior

Recovery from Elective Herniorrhaphy

Description

A data set on measures of post-operative recovery of 32 patients undergoing an elective herniorrhaphy operation, in relation to pre-operative measures.

Usage

```
data(Hernior)
```

Format

A data frame with 32 observations on the following 9 variables.

age patient age

sex patient sex, a factor with levels f m

pstat physical status (ignoring that associated with the operation). A 1-5 scale, with 1=perfect health, 5=very poor health.

build body build, a 1-5 scale, with 1=emaciated, 2=thin, 3=average, 4=fat, 5=obese.

cardiac, resp preoperative complications with heart and respiration, 1-4 scales, with 1=none, 2=mild, 3=moderate, 4=severe.

leave condition upon leaving the recovery room, a 1-4 scale, with 1=routine recovery, 2=intensive care for observation overnight, 3=intensive care, with moderate care required, 4=intensive care, with moderate care required.

los length of stay in hospital after operation (days)

nurse level of nursing required one week after operation, a 1-5 scale, with 1=intense, 2=heavy, 3=moderate, 4=light, 5=none (?); see Details

Details

leave, nurse and los are outcome measures; the remaining variables are potential predictors of recovery status.

The variable nurse is recorded as 1-4, with remaining (20) entries entered as "-" in both sources. It is not clear whether this means "none" or NA. The former interpretation was used in constructing the R data frame, so nurse==5 for these observations. Using `Hernior$nurse[Hernior$nurse==5] <- NA` would change to the other interpretation, but render nurse useless in a multivariate analysis.

The ordinal predictors could instead be treated as factors, and there are also potential interactions to be explored.

Source

Mosteller, F. and Tukey, J. W. (1977), *Data analysis and regression*, Reading, MA: Addison-Wesley. Exhibit 8, 567-568. Their source: A study by B. McPeck and J. P. Gilbert of the Harvard Anesthesia Center.

References

Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J. and Ostrowski, E. (1994), *A Handbook of Small Data Sets*, Number 484, 390-391.

Examples

```
str(Hernior)
mod <- lm(cbind(leave, nurse, los) ~ age + sex + pstat + build + cardiac, data=Hernior)
Anova(mod, test="Roy") # actually, all tests are identical

clr <- c("red", "darkgray", "blue", "darkgreen", "magenta", "brown", "black")
heplot(mod, col=clr)
pairs(mod, col=clr)

# Add ellipse to test all 5 regressors simultaneously
hyp <- list("Regr" = c("age", "sexm", "pstat", "build", "cardiac"))
pairs(mod, hypotheses=hyp, col=clr)
```

mark.H0

Mark a point null hypothesis in an HE plot

Description

A utility function to draw and label a point in a 2D (or 3D) HE plot corresponding to a point null hypothesis being tested. This is most useful for repeated measure designs where null hypotheses for within-S effects often correspond to (0,0).

Usage

```
mark.H0(x = 0, y = 0, z=NULL, label, cex = 2, pch = 19, col = "green3", lty = 2, pos = 2)
```

Arguments

x	Horizontal coordinate for H0
y	Vertical coordinate for H0
z	z coordinate for H0. If not NULL, the function assumes that a heplot3d plot has been drawn.
label	Text used to label the point. Defaults to expression(H[0]) in 2D plots.
cex	Point and text size. For 3D plots, the function uses <code>size=5*cex</code> in a call to points3d .
pch	Plot character. Ignored for 3D plots.
col	Color for text, character and lines
lty	Line type for vertical and horizontal reference lines. Not drawn if <code>lty=0</code> .
pos	Position of text. Ignored for 3D plots

Value

None. Used for side effect of drawing on the current plot.

Author(s)

Michael Friendly

See Also

[cross3d](#), ~~~

Examples

```
Vocab.mod <- lm(cbind(grade8,grade9,grade10,grade11) ~ 1, data=VocabGrowth)
idata <-data.frame(grade=ordered(8:11))

heplot(Vocab.mod, type="III", idata=idata, idesign=~grade, iterm="grade",
main="HE plot for Grade effect")
mark.H0()
```

MockJury

Effects Of Physical Attractiveness Upon Mock Jury Decisions

Description

Male participants were shown a picture of one of three young women. Pilot work had indicated that the one woman was beautiful, another of average physical attractiveness, and the third unattractive. Participants rated the woman they saw on each of twelve attributes. These measures were used to check on the manipulation by the photo.

Then the participants were told that the person in the photo had committed a Crime, and asked to rate the seriousness of the crime and recommend a prison sentence, in Years.

Does attractiveness of the "defendent" influence the sentence or perceived seriousness of the crime?
Does attractiveness interact with the nature of the crime?

Usage

```
data(MockJury)
```

Format

A data frame with 114 observations on the following 17 variables.

Attr Attractiveness of the photo, a factor with levels Beautiful Average Unattractive

Crime Type of crime, a factor with levels Burglary (theft of items from victim's room) Swindle (conned a male victim)

Years length of sentence given the defendant by the mock juror subject

Serious a rating of how serious the subject thought the defendant's crime was

exciting rating of the photo for 'exciting'
 calm rating of the photo for 'calm'
 independent rating of the photo for 'independent'
 sincere rating of the photo for 'sincere'
 warm rating of the photo for 'warm'
 phyattr rating of the photo for 'physical attractiveness'
 sociable rating of the photo for 'exciting'
 kind rating of the photo for 'kind'
 intelligent rating of the photo for 'intelligent'
 strong rating of the photo for 'strong'
 sophisticated rating of the photo for 'sophisticated'
 happy rating of the photo for 'happy'
 ownPA self-rating of the subject for 'physical attractiveness'

Source

From Dr. Wuensch's StatData Page, <http://core.ecu.edu/psyc/wuenschk/StatData/PLASTER.dat>

References

Data from the thesis by Plaster, M. E. (1989). *Inmates as mock jurors: The effects of physical attractiveness upon juridic decisions*. M.A. thesis, Greenville, NC: East Carolina University.

Examples

```
# manipulation check: test ratings of the photos classified by Attractiveness
jury.mod1 <- lm( cbind(phyattr, happy, independent, sophisticated) ~ Attr, data=MockJury)
Anova(jury.mod1, test="Roy")
##
## Type II MANOVA Tests: Roy test statistic
##      Df test stat approx F num Df den Df    Pr(>F)
## Attr 2    1.7672  48.156      4    109 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

heplot(jury.mod1, main="HE plot for manipulation check")
pairs(jury.mod1)

if (require(candisc)) {
  jury.can <- candisc(jury.mod1)
  jury.can
  heplot(jury.can, main="Canonical HE plot")
}

# influence of Attr of photo and nature of crime on Serious and Years
jury.mod2 <- lm( cbind(Serious, Years) ~ Attr * Crime, data=MockJury)
```

```

Anova(jury.mod2, test="Roy")
heplot(jury.mod2)

# stepdown test (ANCOVA), controlling for Serious
jury.mod3 <- lm( Years ~ Serious + Attr * Crime, data=MockJury)
Anova(jury.mod3)

# need to consider heterogeneous slopes?
jury.mod4 <- lm( Years ~ Serious * Attr * Crime, data=MockJury)
anova(jury.mod3, jury.mod4)

```

pairs.mlm

Pairwise HE Plots

Description

The function (in the form of an `mlm` method for the generic `pairs` function) constructs a “matrix” of pairwise HE plots (see [heplot](#)) for a multivariate linear model.

Usage

```

## S3 method for class 'mlm'
pairs(x, variables, var.labels, type = c("II", "III", "2", "3"),
      idata=NULL, idesign=NULL, icontrasts=NULL, imatrix=NULL, iterm=NULL, manova,
      offset.axes = 0.05, digits = getOption("digits") - 1, fill=FALSE, fill.alpha=0.3, ...)

```

Arguments

<code>x</code>	an object of class <code>mlm</code> .
<code>variables</code>	indices or names of the three or more response variables to be plotted; defaults to all of the responses.
<code>var.labels</code>	labels for the variables plotted in the diagonal panels; defaults to names of the response variables.
<code>type</code>	“type” of sum-of-squares-and-products matrices to compute; one of “II”, “III”, “2”, or “3”, where “II” is the default (and “2” is a synonym).
<code>idata</code>	an optional data frame giving a factor or factors defining the intra-subject model for multivariate repeated-measures data. See Details of Anova for an explanation of the intra-subject design and for further explanation of the other arguments relating to intra-subject factors.
<code>idesign</code>	a one-sided model formula using the “data” in <code>idata</code> and specifying the intra-subject design for repeated measure models.
<code>icontrasts</code>	names of contrast-generating functions to be applied by default to factors and ordered factors, respectively, in the within-subject “data”; the contrasts must produce an intra-subject model matrix in which different terms are orthogonal. The default is <code>c("contr.sum", "contr.poly")</code> .

imatrix	In lieu of <code>idata</code> and <code>idesign</code> , you can specify the intra-subject design matrix directly via <code>imatrix</code> , in the form of list of named elements. Each element gives the columns of the within-subject model matrix for an intra-subject term to be tested, and must have as many rows as there are responses; the columns of the within-subject model matrix for <i>different</i> terms must be mutually orthogonal. <i>This functionality requires car version 2.0 or later.</i>
iterm	For repeated measures designs, you must specify one intra-subject term (a character string) to select the SSPE (E) matrix used in the HE plot. Hypothesis terms plotted include the <code>iterm</code> effect as well as all interactions of <code>iterm</code> with terms.
manova	optional <code>Anova.mlm</code> object for the model; if absent a MANOVA is computed. Specifying the argument can therefore save computation in repeated calls.
offset.axes	proportion to extend the axes in each direction; defaults to 0.05.
digits	number of significant digits in axis end-labels; taken from the "digits" option.
fill	A logical vector indicating whether each ellipse should be filled or not. The first value is used for the error ellipse, the rest — possibly recycled — for the hypothesis ellipses; a single fill value can be given. Defaults to FALSE for backward compatibility. See Details of heplot
fill.alpha	Alpha transparency for filled ellipses, a numeric scalar or vector of values within [0, 1], where 0 means fully transparent and 1 means fully opaque. Defaults to 0.3.
...	arguments to pass down to <code>heplot</code> , which is used to draw each panel of the display.

References

- Friendly, M. (2006). Data Ellipses, HE Plots and Reduced-Rank Displays for Multivariate Linear Models: SAS Software and Examples *Journal of Statistical Software*, 17(6), 1-42. <http://www.jstatsoft.org/v17/i06/>
- Friendly, M. (2007). HE plots for Multivariate General Linear Models. *Journal of Computational and Graphical Statistics*, 16(2) 421-444. <http://datavis.ca/papers/jcgs-heplots.pdf>

See Also

[heplot](#), [heplot3d](#)

Examples

```
# ANCOVA, assuming equal slopes
rohwer.mod <- lm(cbind(SAT, PPVT, Raven) ~ SES + n + s + ns + na + ss, data=Rohwer)

# View all pairs, with ellipse for all 5 regressors
pairs(rohwer.mod, hypotheses=list("Regr" = c("n", "s", "ns", "na", "ss")))
```

Plastic

Plastic Film Data

Description

An experiment was conducted to determine the optimum conditions for extruding plastic film. Three responses were measured in relation to two factors, rate of extrusion and amount of an additive.

Usage

```
data(Plastic)
```

Format

A data frame with 20 observations on the following 5 variables.

tear a numeric vector: tear resistance

gloss a numeric vector: film gloss

opacity a numeric vector: film opacity

rate a factor representing change in the rate of extrusion with levels Low (-10%), High (10%)

additive a factor with levels Low (1.0%), High (1.5%)

Source

Johnson, R.A. & Wichern, D.W. (1992). *Applied Multivariate Statistical Analysis*, 3rd ed., Prentice-Hall. Example 6.12 (p. 266).

References

Krzanowski, W. J. (1988). *Principles of Multivariate Analysis. A User's Perspective*. Oxford. (p. 381)

Examples

```
str(Plastic)
plastic.mod <- lm(cbind(tear, gloss, opacity) ~ rate*additive, data=Plastic)
Anova(plastic.mod)

pairs(plastic.mod)
```

RatWeight

Weight Gain in Rats Exposed to Thiouracil and Thyroxin

Description

The data are from a study of weight gain, where investigators randomly assigned 30 rats to three treatment groups: treatment 1 was a control (no additive); treatments 2 and 3 consisted of two different additives (thiouracil and thyroxin respectively) to the rats drinking water. Weight was measured at baseline (week 0) and at weeks 1, 2, 3, and 4. Due to an accident at the beginning of the study, data on 3 rats from the thyroxin group are unavailable.

Usage

```
data(RatWeight)
```

Format

A data frame with 27 observations on the following 6 variables.

trt a factor with levels Control Thiouracil Thyroxin

wt0 Weight at Week 0 (baseline weight)

wt1 Weight at Week 1

wt2 Weight at Week 2

wt3 Weight at Week 3

wt4 Weight at Week 4

Details

The trt factor comes supplied with contrasts comparing Control to each of Thiouracil and Thyroxin.

Source

Originally from Box (1950), Table D (page 389), where the values for weeks 1-4 were recorded as the gain in weight for that week.

Fitzmaurice, G. M. and Laird, N. M. and Ware, J. H (2004). *Applied Longitudinal Analysis*, New York, NY: Wiley-Interscience. <http://biosun1.harvard.edu/~fitzmaur/ala/rat.txt>.

References

Box, G.E.P. (1950). Problems in the analysis of growth and wear curves. *Biometrics*, 6, 362-389.

Friendly, Michael (2010). HE Plots for Repeated Measures Designs. *Journal of Statistical Software*, 37(4), 1-40. URL <http://www.jstatsoft.org/v37/i04/>.

Examples

```

data(RatWeight)
contrasts(RatWeight$trt)

rat.mod <- lm(cbind(wt0, wt1, wt2, wt3, wt4) ~ trt, data=RatWeight)
rat.mod

idata <- data.frame(week = ordered(0:4))
Anova(rat.mod, idata=idata, idesign=~week, test="Roy")

# quick look at between group effects
pairs(rat.mod)

# between-S, baseline & week 4
heplot(rat.mod, col=c("red", "blue", "green3", "green3"),
variables=c(1,5),
hypotheses=c("trt1", "trt2"),
main="Rat weight data, Between-S effects")

# within-S
heplot(rat.mod, idata=idata, idesign=~week, iterm="week",
col=c("red", "blue", "green3"),
# hypotheses=c("trt1", "trt2"),
main="Rat weight data, Within-S effects")

```

ReactTime

Reaction Time Data

Description

Data from Maxwell and Delaney (1990, p. 497) representing the reaction times of 10 subjects in some task where visual stimuli are tilted at 0, 4, and 8 degrees; with noise absent or present. Each subject responded to 3 tilt x 2 noise = 6 conditions. The data thus comprise a repeated measure design with two within-S factors.

Usage

```
data(ReactTime)
```

Format

A data frame with 10 observations giving the reaction time for the 6 conditions.

```

deg0NA a numeric vector
deg4NA a numeric vector
deg8NA a numeric vector
deg0NP a numeric vector
deg4NP a numeric vector
deg8NP a numeric vector

```

Source

Baron, J. and Li, Y. (2003). *Notes on the use of R for psychology experiments and questionnaires*, <http://cran.r-project.org/doc/contrib/Baron-rpsych.pdf>

References

Michael Friendly (2010). HE Plots for Repeated Measures Designs. *Journal of Statistical Software*, 37(4), 1-40. URL <http://www.jstatsoft.org/v37/i04/>.

Maxwell, S. E. & Delaney, H. D. (1990). *Designing Experiments and Analyzing Data: A model comparison perspective*. Pacific Grove, CA: Brooks/Cole.

Examples

```
data(ReactTime)
(RT.mod <- lm(as.matrix(ReactTime)~1))

# within-S factors
within <- expand.grid(tilt=ordered(c(0,4,8)), noise=c("NA", "NP"))
Anova(RT.mod, idata=within, idesign=~tilt * noise)

heplot(RT.mod, idata=within, idesign=~tilt * noise, iterm="tilt")

# plotting means and std errors directly

levels <- expand.grid(Tilt=c(0,4,8), noise=c("NA", "NP"))
(means.df <- data.frame(levels, mean=colMeans(ReactTime), se=sqrt(diag(var(ReactTime)))/9))

with(means.df, {
  plot(Tilt, mean, type="n", main="Reaction Time data", xlab="Tilt", ylab="Reaction time")
  colors <- rep(c("red", "blue"), each=3)
  pts <- rep(c(15, 16), each=3)

  lines(Tilt[1:3], mean[1:3], col="red", lwd=2)
  lines(Tilt[4:6], mean[4:6], col="blue", lwd=2)
  points(Tilt, mean, pch=pts, col=colors, cex=1.2)
  arrows(Tilt, mean-se, Tilt, mean+se, angle=90, code=3,
  col=colors, len=.05, lwd=2)
  # labels at last point, in lieu of legend
  text(Tilt[3], mean[3]-10, labels="NA", col="red", pos=1)
  text(Tilt[6], mean[6]-10, labels="NP", col="blue", pos=1)
}
)
```

Description

Data from an experiment by William D. Rohwer on kindergarten children designed to examine how well performance on a set of paired-associate (PA) tasks can predict performance on some measures of aptitude and achievement.

Usage

```
data(Rohwer)
```

Format

A data frame with 69 observations on the following 10 variables.

group a numeric vector, corresponding to SES
 SES Socioeconomic status, a factor with levels Hi Lo
 SAT a numeric vector: score on a Student Achievement Test
 PPVT a numeric vector: score on the Peabody Picture Vocabulary Test
 Raven a numeric vector: score on the Raven Progressive Matrices Test
 n a numeric vector: performance on a 'named' PA task
 s a numeric vector: performance on a 'still' PA task
 ns a numeric vector: performance on a 'named still' PA task
 na a numeric vector: performance on a 'named action' PA task
 ss a numeric vector: performance on a 'sentence still' PA task

Details

The variables SAT, PPVT and Raven are responses to be potentially explained by performance on the paired-associate (PA) learning taskn, s, ns, na, and ss.

Source

Timm, N.H. 1975). *Multivariate Analysis with Applications in Education and Psychology*. Wadsworth (Brooks/Cole), Examples 4.3 (p. 281), 4.7 (p. 313), 4.13 (p. 344).

References

Friendly, M. (2007). HE plots for Multivariate General Linear Models. *Journal of Computational and Graphical Statistics*, **16**(2) 421–444. <http://datavis.ca/papers/jcgs-heplots.pdf>

Examples

```
str(Rohwer)

## ANCOVA, assuming equal slopes
rohwer.mod <- lm(cbind(SAT, PPVT, Raven) ~ SES + n + s + ns + na + ss, data=Rohwer)
Anova(rohwer.mod)
```

```

# Visualize the ANCOVA model
heplot(rohwer.mod)
# Add ellipse to test all 5 regressors
heplot(rohwer.mod, hypotheses=list("Regr" = c("n", "s", "ns", "na", "ss")))
# View all pairs
pairs(rohwer.mod, hypotheses=list("Regr" = c("n", "s", "ns", "na", "ss")))

# or 3D plot
col <- c("pink", "black", "blue", "cyan", "magenta", "brown", "gray")
heplot3d(rohwer.mod, hypotheses=list("Regr" = c("n", "s", "ns", "na", "ss")), col=col)

## fit separate, independent models for Lo/Hi SES
rohwer.ses1 <- lm(cbind(SAT, PPVT, Raven) ~ n + s + ns + na + ss, data=Rohwer, subset=SES=="Hi")
rohwer.ses2 <- lm(cbind(SAT, PPVT, Raven) ~ n + s + ns + na + ss, data=Rohwer, subset=SES=="Lo")

# overlay the separate HE plots
heplot(rohwer.ses1, ylim=c(40,110),col=c("red", "black"))
heplot(rohwer.ses2, add=TRUE, col=c("blue", "black"), grand.mean=TRUE, error.ellipse=TRUE)

```

RootStock

Growth of Apple Trees from Different Root Stocks

Description

In a classic experiment carried out from 1918 to 1934, growth of apple trees of six different rootstocks were compared on four measures of size.

Usage

```
data(RootStock)
```

Format

A data frame with 48 observations on the following 5 variables.

rootstock a factor with levels 1 2 3 4 5 6

girth4 a numeric vector: trunk girth at 4 years (mm x 100)

ext4 a numeric vector: extension growth at 4 years (m)

girth15 a numeric vector: trunk girth at 15 years (mm x 100)

weight15 a numeric vector: weight of tree above ground at 15 years (lb x 1000)

Details

This is a balanced, one-way MANOVA design, with n=8 trees for each rootstock.

Source

Andrews, D. and Herzberg, A. (1985). *Data: A Collection of Problems from Many Fields for the Student and Research Worker* Springer-Verlag, pp. 357–360.

References

Rencher, A. C. (1995). *Methods of Multivariate Analysis*. New York: Wiley, Table 6.2

Examples

```
data(RootStock)
## maybe str(RootStock) ; plot(RootStock) ...
root.mod <- lm(cbind(girth4, ext4, girth15, weight15) ~ rootstock, data=RootStock)
Anova(root.mod)

pairs(root.mod)

# test two orthogonal contrasts among the rootstocks
hyp <- matrix(c(2,-1,-1,-1,-1,2, 1, 0,0,0,0,-1), 2, 6, byrow=TRUE)
linearHypothesis(root.mod, hyp)
heplot(root.mod, hypotheses=list(Contrasts=hyp, C1=hyp[1,], C2=hyp[2,]))

heplot1d(root.mod, hypotheses=list(Contrasts=hyp, C1=hyp[1,], C2=hyp[2,]))
```

SocGrades

Grades in a Sociology Course

Description

The data set SocGrades contains four outcome measures on student performance in an introductory sociology course together with six potential predictors. These data were used by Marascuilo and Levin (1983) for an example of canonical correlation analysis, but are also suitable as examples of multivariate multiple regression, MANOVA, MANCOVA and step-down analysis in multivariate linear models.

Usage

```
data(SocGrades)
```

Format

A data frame with 40 observations on the following 10 variables.

class Social class, an ordered factor with levels 1 > 2 > 3

sex sex, a factor with levels F M

gpa grade point average

boards College Board test scores
 hssoc previous high school unit in sociology, a factor with 2 no, yes
 pretest score on course pretest
 midterm1 score on first midterm exam
 midterm2 score on second midterm exam
 final score on final exam
 eval course evaluation

Details

The factors `class`, `sex`, and `hssoc` can be used with `as.numeric` in correlational analyses.

Source

Marascuilo, L. A. and Levin, J. R. (1983). *Multivariate Statistics in the Social Sciences* Monterey, CA: Brooks/Cole, Table 5-1, p. 192.

Examples

```
data(SocGrades)
# basic MLM
grades.mod <- lm(cbind(midterm1, midterm2, final, eval) ~
class + sex + gpa + boards + hssoc + pretest, data=SocGrades)

Anova(grades.mod, test="Roy")

clr <- c("red", "blue", "darkgreen", "magenta", "brown", "black", "darkgray")
heplot(grades.mod, col=clr)
pairs(grades.mod, col=clr)

heplot3d(grades.mod, col=clr, wire=FALSE)
```

statList

Calculate statistics for levels of factors

Description

`statList` provides a general method for calculating univariate or multivariate statistics for a matrix or `data.frame` stratified by one or more factors.

Usage

```
statList(X, factors, FUN, drop = FALSE, ...)

colMeansList(X, factors, drop = FALSE, ...)

covList(X, factors, drop = FALSE, ...)
```

Arguments

<code>X</code>	A matrix or data frame containing the variables to be summarized
<code>factors</code>	A vector, matrix or data frame containing the factors for which <code>X</code> is to be summarized. If <code>factors</code> is not specified, the result is calculated for all of the data in <code>X</code> .
<code>FUN</code>	A function to be applied to the pieces of <code>X</code> , as split by <code>factors</code> .
<code>drop</code>	Logical, indicating whether empty levels of <code>factors</code> are to be dropped from the result.
<code>...</code>	Other arguments, passed to <code>FUN</code> .

Details

`statList` is the general function. `X` is first split by `factors`, and `FUN` is applied to the result. `colMeansList` and `covList` are just calls to `statList` with the appropriate `FUN`.

Value

Returns a list of items corresponding to the unique elements in `factors`, or the interaction of `factors`. Each item is the result of applying `FUN` to that collection of rows of `X`. The items are named according to the levels in `factors`.

Author(s)

Michael Friendly

See Also

[colMeans](#), [termMeans](#)

Examples

```
# grand means
statList(iris[,1:4], FUN=colMeans)
# species means
statList(iris[,1:4], iris$Species, FUN=colMeans)
# same
colMeansList(iris[,1:4], iris$Species)

# var-cov matrices, by species
covList(iris[,1:4], iris$Species)

# multiple factors
iris$Dummy <- sample(c("Hi", "Lo"), 150, replace=TRUE)
colMeansList(iris[,1:4], iris[,5:6])
```

`termMeans`*Calculate Means for a Term in a Multivariate Linear Model*

Description

`termMeans` is a utility function designed to calculate means for the levels of factor(s) for any term in a multivariate linear model.

Usage

```
termMeans(mod, term, label.factors=FALSE, abbrev.levels=FALSE)
```

Arguments

<code>mod</code>	An <code>mlm</code> model object
<code>term</code>	A character string indicating a given term in the model. All factors in the term must be included in the model, even if they are in the model data frame.
<code>label.factors</code>	If true, the rownames for each row in the result include the name(s) of the factor(s) involved, followed by the level values. Otherwise, the rownames include only the levels of the factor(s), with multiple factors separated by ':'
<code>abbrev.levels</code>	Either a logical or an integer, specifying whether the levels values of the factors in the term are to be abbreviated in constructing the rownames. An integer specifies the minimum length of the abbreviation for each factor in the term.

Value

Returns a matrix whose columns correspond to the response variables in the model and whose rows correspond to the levels of the factor(s) in the term.

Author(s)

Michael Friendly

See Also

[aggregate](#), [colMeans](#)
[statList](#), [colMeansList](#)

Examples

```
factors <- expand.grid(A=factor(1:3),B=factor(1:2),C=factor(1:2))
n <- nrow(factors)
responses <- data.frame(Y1=10+round(10*rnorm(n)),Y2=10+round(10*rnorm(n)))
test <- data.frame(factors, responses)
mod <- lm(cbind(Y1,Y2) ~ A*B, data=test)

termMeans(mod, "A")
```

```

termMeans(mod, "A:B")
termMeans(mod, "A:B", label.factors=TRUE)
## Not run:
termMeans(mod, "A:B:C")    # generates an error

## End(Not run)

plastic.mod <- lm(cbind(tear, gloss, opacity) ~ rate*additive, data=Plastic)
colors = c("red", "darkblue", "darkgreen", "brown")
heplot(plastic.mod, col=colors, cex=1.25)
# add means for interaction term
intMeans <- termMeans(plastic.mod, 'rate:additive', abbrev=2)
points(intMeans[,1], intMeans[,2], pch=18, cex=1.2, col="brown")
text(intMeans[,1], intMeans[,2], rownames(intMeans), adj=c(0.5,1), col="brown")

```

trans.colors

Make Colors Transparent

Description

Takes a vector of colors (as color names or rgb hex values) and adds a specified alpha transparency to each.

Usage

```
trans.colors(col, alpha = 0.5, names = NULL)
```

Arguments

col	A character vector of colors, either as color names or rgb hex values
alpha	alpha transparency value(s) to apply to each color (0 means fully transparent and 1 means opaque)
names	optional character vector of names for the colors

Details

Colors (col) and alpha need not be of the same length. The shorter one is replicated to make them of the same length.

Value

A vector of color values of the form "#rrggbbaa"

Author(s)

Michael Friendly

See Also

[col2rgb](#), [rgb](#),

Examples

```
trans.colors(palette(), alpha=0.5)

# alpha can be vectorized
trans.colors(palette(), alpha=seq(0, 1, length=length(palette()))))

# lengths need not match: shorter one is repeated as necessary
trans.colors(palette(), alpha=c(.1, .2))

trans.colors(colors()[1:20])

# single color, with various alphas
trans.colors("red", alpha=seq(0,1, length=5))
# assign names
trans.colors("red", alpha=seq(0,1, length=5), names=paste("red", 1:5, sep=""))
```

VocabGrowth

Vocabulary growth data

Description

Data from the Laboratory School of the University of Chicago. They consist of scores from a cohort of pupils in grades 8-11 on the vocabulary section of the Cooperative Reading Test. The scores are scaled to a common, but arbitrary origin and unit of measurement, so as to be comparable over the four grades.

Usage

```
data(VocabGrowth)
```

Format

A data frame with 64 observations on the following 4 variables.

grade8 Grade 8 vocabulary score
grade9 Grade 9 vocabulary score
grade10 Grade 10 vocabulary score
grade11 Grade 11 vocabulary score

Details

Since these data cover an age range in which physical growth is beginning to decelerate, it is of interest whether a similar effect occurs in the acquisition of new vocabulary.

Source

R.D. Bock, *Multivariate statistical methods in behavioral research*, McGraw-Hill, New York, 1975, pp453.

References

Friendly, Michael (2010). HE Plots for Repeated Measures Designs. *Journal of Statistical Software*, 37(4), 1-40. URL <http://www.jstatsoft.org/v37/i04/>.

Keesling, J.W., Bock, R.D. et al, "The Laboratory School study of vocabulary growth", University of Chicago, 1975.

Examples

```
data(VocabGrowth)

# Standard Multivariate & Univariate repeated measures analysis
Vocab.mod <- lm(cbind(grade8,grade9,grade10,grade11) ~ 1, data=VocabGrowth)
idata <- data.frame(grade=ordered(8:11))
Anova(Vocab.mod, idata=idata, idesign=~grade, type="III")

##Type III Repeated Measures MANOVA Tests: Pillai test statistic
##          Df test stat approx F num Df den Df   Pr(>F)
##(Intercept) 1    0.653  118.498     1    63 4.115e-16 ***
##grade       1    0.826   96.376     3    61 < 2.2e-16 ***

heplot(Vocab.mod, type="III", idata=idata, idesign=~grade, item="grade",
main="HE plot for Grade effect")

### doing this 'manually' by explicitly transforming Y -> Y M
# calculate Y M, using polynomial contrasts
trends <- as.matrix(VocabGrowth) %>% poly(8:11, degree=3)
colnames(trends)<- c("Linear", "Quad", "Cubic")

# test all trend means = 0 == Grade effect
within.mod <- lm(trends ~ 1)

Manova(within.mod)
heplot(within.mod, terms="(Intercept)", col=c("red", "blue"), type="3",
term.labels="Grade",
main="HE plot for Grade effect")
mark.H0()
```

WeightLoss

Weight Loss Data

Description

Contrived data on weight loss and self esteem over three months, for three groups of individuals: Control, Diet and Diet + Exercise. The data constitute a double-multivariate design.

Usage

```
data(WeightLoss)
```

Format

A data frame with 34 observations on the following 7 variables.

group a factor with levels Control Diet DietEx.

wl1 Weight loss at 1 month

wl2 Weight loss at 2 months

wl3 Weight loss at 3 months

se1 Self esteem at 1 month

se2 Self esteem at 2 months

se3 Self esteem at 3 months

Details

Helmert contrasts are assigned to group, comparing Control vs. (Diet DietEx) and Diet vs. DietEx.

Source

Originally taken from <http://www.csun.edu/~ata20315/psy524/main.htm>, but modified slightly

References

Friendly, Michael (2010). HE Plots for Repeated Measures Designs. *Journal of Statistical Software*, 37(4), 1-40. URL <http://www.jstatsoft.org/v37/i04/>.

Examples

```
data(WeightLoss)
str(WeightLoss)
table(WeightLoss$group)

contrasts(WeightLoss$group) <- matrix(c(-2,1,1, 0, -1, 1),ncol=2)
(wl.mod<-lm(cbind(wl1,wl2,wl3,se1,se2,se3)~group, data=WeightLoss))

heplot(wl.mod, hypotheses=c("group1", "group2"))
pairs(wl.mod, variables=1:3)
pairs(wl.mod, variables=4:6)

# within-S variables
within <- data.frame(measure=rep(c("Weight loss", "Self esteem"),each=3), month=rep(ordered(1:3),2))

# doubly-multivariate analysis: requires car 2.0+
## Not run:
if (packageDescription("car")["Version"] >= 2) {
  imatrix <- matrix(c(
```

```

1,0,-1, 1, 0, 0,
1,0, 0,-2, 0, 0,
1,0, 1, 1, 0, 0,
0,1, 0, 0,-1, 1,
0,1, 0, 0, 0,-2,
0,1, 0, 0, 1, 1), 6, 6, byrow=TRUE)
# NB: for heplots the columns of imatrix should have names
colnames(imatrix) <- c("WL", "SE", "WL.L", "WL.Q", "SE.L", "SE.Q")
rownames(imatrix) <- colnames(WeightLoss)[-1]
(imatrix <- list(measure=imatrix[,1:2], month=imatrix[,3:6]))
contrasts(WeightLoss$group) <- matrix(c(-2,1,1, 0,-1,1), ncol=2)
(wl.mod<-lm(cbind(wl1, wl2, wl3, se1, se2, se3)~group, data=WeightLoss))
(wl.aov <- Anova(wl.mod, imatrix=imatrix, test="Roy"))

heplot(wl.mod, imatrix=imatrix, iterm="group:measure")
}

## End(Not run)

# do the correct analysis 'manually'
unit <- function(n, prefix="") {
J <-matrix(rep(1, n), ncol=1)
rownames(J) <- paste(prefix, 1:n, sep="")
J
}

measure <- kronecker(diag(2), unit(3, 'M')/3, make.dimnames=TRUE)
colnames(measure)<- c('WL', 'SE')

between <- as.matrix(WeightLoss[,-1])

between.mod <- lm(between ~ group, data=WeightLoss)
Anova(between.mod)

heplot(between.mod, hypotheses=c("group1", "group2"),
xlab="Weight Loss", ylab="Self Esteem",
col=c("red", "blue", "brown"),
main="Weight Loss & Self Esteem: Group Effect")

month <- kronecker(diag(2), poly(1:3), make.dimnames=TRUE)
colnames(month)<- c('WL', 'SE')
trends <- as.matrix(WeightLoss[,-1])
within.mod <- lm(trends ~ group, data=WeightLoss)
Anova(within.mod)

heplot(within.mod)
heplot(within.mod, hypotheses=c("group1", "group2"),
xlab="Weight Loss", ylab="Self Esteem",
type="III", remove.intercept=FALSE,
term.labels=c("month", "group:month"),
main="Weight Loss & Self Esteem: Within-S Effects")
mark.H0()

```

Index

*Topic **aplot**

- arrow3d, 6
- cross3d, 9
- ellipse3d.axes, 10
- heplot, 18
- heplot1d, 23
- heplot3d, 26
- heplots-package, 2
- mark.H0, 31

*Topic **color**

- trans.colors, 46

*Topic **datasets**

- Adopted, 5
- Bees, 7
- FootHead, 13
- Headache, 16
- Hernior, 30
- MockJury, 32
- Plastic, 36
- RatWeight, 37
- ReactTime, 38
- Rohwer, 39
- RootStock, 41
- SocGrades, 42
- VocabGrowth, 47
- WeightLoss, 48

*Topic **dynamic**

- cross3d, 9
- ellipse3d.axes, 10
- heplot3d, 26

*Topic **hplot**

- heplot, 18
- heplot1d, 23
- heplot3d, 26
- heplots-package, 2
- pairs.mlm, 34

*Topic **manip**

- gsorth, 14
- termMeans, 45

*Topic **models**

- termMeans, 45

*Topic **multivariate**

- etasq, 11
- heplot, 18
- heplot1d, 23
- heplot3d, 26
- heplots-package, 2
- pairs.mlm, 34
- statList, 43

*Topic **package**

- heplots-package, 2

*Topic **utilities**

- statList, 43

Adopted, 5

aggregate, 45

Anova, 4, 5, 12, 19, 22, 24, 25, 27, 29, 34

arrow3d, 6

Bees, 7

bg, 28

candisc, 21, 22

candisc-package, 4, 5

car-package, 4

col2rgb, 47

colMeans, 44, 45

colMeansList, 45

colMeansList(statList), 43

covList(statList), 43

cross3d, 9, 32

ellipse3d, 11

ellipse3d.axes, 10

etasq, 11

ex1605, 5, 6

FootHead, 13

gsorth, 14

Headache, [16](#)
heplot, [4](#), [18](#), [25](#), [29](#), [34](#), [35](#)
heplot.candisc, [21](#), [22](#)
heplot1d, [4](#), [22](#), [23](#)
heplot3d, [4](#), [22](#), [25](#), [26](#), [35](#)
heplot3d.candisc, [21](#), [29](#)
heplots (heplots-package), [2](#)
heplots-package, [2](#)
Hernior, [30](#)

linear.hypothesis, [18](#), [21](#)
linearHypothesis, [4](#), [5](#), [22](#), [23](#), [25](#), [26](#), [29](#)
lines3d, [7](#)

manova, [5](#)
mark.H0, [19](#), [22](#), [31](#)
material3d, [7](#)
MockJury, [32](#)
movie3d, [29](#)

pairs, [34](#)
pairs.mlm, [4](#), [22](#), [25](#), [29](#), [34](#)
Plastic, [36](#)
play3d, [29](#)
points3d, [31](#)

qr, [15](#)

RatWeight, [37](#)
ReactTime, [38](#)
rgb, [47](#)
rgl-package, [29](#)
Rohwer, [39](#)
RootStock, [41](#)

segments3d, [7](#), [9–11](#)
shade3d, [28](#)
SocGrades, [42](#)
statList, [43](#), [45](#)

termMeans, [44](#), [45](#)
text3d, [11](#)
trans.colors, [22](#), [46](#)

viewpoint, [28](#)
VocabGrowth, [47](#)

WeightLoss, [48](#)
wire3d, [28](#)