Package 'heplots'

October 4, 2026

Type Package

Title Visualizing Hypothesis Tests in Multivariate Linear Models

Version 1.8.6

Date 2026-10-04

Maintainer Michael Friendly <friendly@yorku.ca>

Description Provides HE plot and other functions for visualizing hypothesis tests in multivariate linear models. HE plots represent sums-of-squares-and-products matrices for linear hypotheses and for error using ellipses (in two dimensions) and ellipsoids (in three dimensions). It also provides other tools for analysis and graphical display of the models such as robust methods and homogeneity of variance covariance matrices. The related 'candisc' package provides visualizations in a reduced-rank canonical discriminant space when there are more than a few response variables.

Language en-US

Depends R (>= 4.1.0)

Imports car, generics, MASS, graphics, grDevices, stats, magrittr, purrr, rgl, tibble, boot, glue

Suggests broom, candisc, carData, lmtest, effects, reshape, gplots, nlme, lattice, reshape2, corrgram, animation, mvinfluence, knitr, litedown, rmarkdown, markdown, dplyr, tidyr, ggplot2, bookdown, patchwork, tinytable, here, htmltools, Sleuth2, rrcov, archdata, qqtest, ggbiplot, vcdExtra, KernSmooth, aplpack, foreign, robustbase, effectsize, parameters

LazyLoad yes

LazyData yes

BugReports https://github.com/friendly/heplots/issues

License GPL (>= 2)

URL https://friendly.github.io/heplots/, https://github.com/friendly/heplots

Encoding UTF-8

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Michael Friendly [aut, cre] (ORCID:
<https://orcid.org/0000-0002-3237-0941>),
John Fox [aut] (ORCID: <https://orcid.org/0000-0002-1196-8012>),
Georges Monette [aut] (ORCID: <https://orcid.org/0000-0003-0076-5532>),
Phil Chalmers [ctb] (ORCID: <https://orcid.org/0000-0001-5332-2810>),
Duncan Murdoch [ctb],
Gavin Klorfine [ctb]

Repository CRAN

Date/Publication 2026-10-04 15:00:02 UTC

Contents

heplots-package 4
AddHealth 6
Adopted 7
arrow3d 9
bartlettTests 10
bbox3d 11
Bees 12
boxM 14
coefplot 17
colDevs 20
covEllipses 22
cqplot 26
cross3d 30
df.terms 31
Diabetes 31
distancePlot 33
dogfood 36
eigstatCI 38
ellipse.axes 40
ellipse.box 42
ellipse3d.axes 44
Ellipsoid 45
etasq 47
FootHead 49
glance.mlm 50
gsorth 51
Headache 53
heplot 54
heplot1d 61
heplot3d 65

Hernior . . . 71
interpPlot . . . 73
Iwasaki_Big_Five . . . 77
label.ellipse . . . 78
LearnDis . . . 81
leveneTests . . . 82
logdetCI . . . 84
Mahalanobis . . . 86
mark.H0 . . . 87
mathscore . . . 88
MockJury . . . 89
NeuroCog . . . 91
NLSY . . . 93
noteworthy . . . 94
oral . . . 96
Oslo . . . 98
Overdose . . . 99
pairs.mlm . . . 101
Parenting . . . 103
peng . . . 104
Plastic . . . 106
plot.boxM . . . 107
plot.robmlm . . . 109
plot_boxM_boot . . . 111
Pottery2 . . . 113
Probe . . . 115
pvPlot . . . 117
RatWeight . . . 119
ReactTime . . . 121
ReadingDisability . . . 122
rel_diff . . . 123
robmlm . . . 125
Rohwer . . . 129
RootStock . . . 131
Sake . . . 132
schooldata . . . 134
schoolsites . . . 136
Skulls . . . 137
SocGrades . . . 139
SocialCog . . . 140
statList . . . 142
stdcoef . . . 143
stdmodel . . . 144
termMeans . . . 146
text_usr . . . 147
TIPI . . . 148
traceCI . . . 150
trans.colors . . . 152

uniStats 153
VocabGrowth 154
WeightLoss 155

Index **158**

`heplots-package` *Visualizing Hypothesis 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). HE plots 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.

Details

The basic theory behind HE plots is described by Friendly (2007). See Fox, Friendly and Monette (2007) for a brief introduction; Friendly & Sigal (2016) for a tutorial on these methods; and Friendly, Monette and Fox (2013) for a general discussion of the role of elliptical geometry in statistical understanding.

Other topics now addressed here include robust MLMs, tests for equality of covariance matrices in MLMs, and chi square Q-Q plots for MLMs.

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. See `vignette(package="heplots")`.

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).

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

list(list("heplot3d")) constructs analogous 3D plots for triples of response variables.

list(list("pairs.mlm")) constructs a "matrix" of pairwise HE plots.

list(list("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 **Y** of responses to a matrix **Y M**, where **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 calculates Type II and Type III tests of multivariate linear hypotheses using the `car::Anova()` and `car::linearHypothesis()` functions.

The candisc::candisc-package package provides functions for visualizing effects for MLM model terms in a low-dimensional canonical space that shows the largest hypothesis relative to error variation. The candisc package now also includes related methods for canonical correlation analysis.

The heplots package also contains a large number of multivariate data sets with examples of analyses and graphic displays. Use data(package="heplots") to see the current list.

Author(s)

Michael Friendly, John Fox, 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. % https://www.jstatsoft.org/v17/i06/, doi:10.18637/jss.v017.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, doi:10.1198/106186007X208407

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*. https://socialsciences.mcmaster.ca/jfox/heplots-dsc-paper.pdf

Friendly, M. (2010). HE Plots for Repeated Measures Designs. *Journal of Statistical Software*, 37(4), 1-40. doi:10.18637/jss.v037.i04.

Fox, J., Friendly, M. & Weisberg, S. (2013). Hypothesis Tests for Multivariate Linear Models Using the car Package. *The R Journal*, **5**(1), https://journal.r-project.org/articles/RJ-2013-004/RJ-2013-004.pdf.

Friendly, M., Monette, G. & Fox, J. (2013). Elliptical Insights: Understanding Statistical Methods Through Elliptical Geometry. *Statistical Science*, 2013, **28** (1), 1-39, http://datavis.ca/papers/ellipses.pdf.

Friendly, M. & Sigal, M. (2014). Recent Advances in Visualizing Multivariate Linear Models. *Revista Colombiana de Estadistica*, **37**, 261-283 doi:10.15446/rce.v37n2spe.47934.

Friendly, M. & Sigal, M. (2016). Graphical Methods for Multivariate Linear Models in Psychological Research: An R Tutorial. Submitted for publication.

See Also

[car::Anova()], [car::linearHypothesis()] for Anova.mlm computations and tests

[`candisc::candisc-package`] for reduced-rank views in canonical space

[stats::manova()] for a different approach to testing effects in MANOVA designs

AddHealth *Adolescent Mental Health Data*

Description

This data was taken from the National Longitudinal Study of Adolescent Health. It is a cross-sectional sample of participants from grades 7–12, described and analyzed by Warne (2014).

Format

A data frame with 4344 observations on the following 3 variables.

`grade` an ordered factor with levels `7` < `8` < `9` < `10` < `11` < `12`

`depression` a numeric vector

`anxiety` a numeric vector

Details

`depression` is the response to the question "In the last month, how often did you feel depressed or blue?"

`anxiety` is the response to the question "In the last month, how often did you have trouble relaxing?"

The responses for `depression` and `anxiety` were recorded on a 5-point Likert scale, with categories `0="Never", 1="Rarely", 2="Occasionally", 3="Often", 4="Every day"`

Source

Warne, R. T. (2014). A primer on Multivariate Analysis of Variance (MANOVA) for Behavioral Scientists. *Practical Assessment, Research & Evaluation*, 19 (1).

Examples

```
data(AddHealth)

if(require(dplyr) & require(ggplot2)) {
# find means & std.errors by grade
means <- AddHealth |>
group_by(grade) |>
  summarise(
    n = n(),
    dep_se = sd(depression, na.rm = TRUE) / sqrt(n),
    anx_se = sd(anxiety, na.rm = TRUE) / sqrt(n),
    depression = mean(depression),
    anxiety = mean(anxiety) ) |>
  relocate(depression, anxiety, .after = grade) |>
  print()

# plot means with std.error bars
ggplot(data = means, aes(x = anxiety, y = depression,
```

```
color = grade)) +
  geom_point(size = 3) +
  geom_errorbarh(aes(xmin = anxiety - anx_se,
                     xmax = anxiety + anx_se)) +
  geom_errorbar(aes(ymin = depression - dep_se,
                    ymax = depression + dep_se)) +
  geom_line(aes(group = 1), linewidth = 1.5) +
  geom_label(aes(label = grade),
             nudge_x = -0.015, nudge_y = 0.02) +
  scale_color_discrete(guide = "none") +
  theme_bw(base_size = 15)
}

# fit mlm
AH.mod <- lm(cbind(anxiety, depression) ~ grade, data=AddHealth)

car::Anova(AH.mod)
summary(car::Anova(AH.mod))

heplot(AH.mod, hypotheses="grade.L",
       fill=c(TRUE, FALSE),
       level = 0.4)
```

`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?

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

Details

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?

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 `Sleuth2::ex1605` in the `Sleuth2` package.

References

Friendly, M. (2010). HE Plots for Repeated Measures Designs. *Journal of Statistical Software*, 37(4), 1-40. doi:10.18637/jss.v037.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

`Sleuth2::ex1605`

Examples

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


require(car)
# test overall multivariate regression
print(linearHypothesis(Adopted.mod, c("AMED","BMIQ")), SSP=FALSE)

# show separate linear regressions
op <- par(mfcol=c(2,4), mar=c(4,4,1,1)+.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")))

if(requireNamespace("rgl")){
heplot3d(Adopted.mod, hypotheses=list("Reg"=c("AMED", "BMIQ")),
 col = c("red", "blue", "black", "gray"), wire=FALSE)
```

```
}

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

# within-S plots
heplot(Adopted.mod, idata=Age, idesign=~Age, iterm="Age",
 cex=1.25, cex.lab=1.4, fill=c(FALSE, TRUE),
 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 `rgl::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

`rgl::lines3d()`, `rgl::segments3d()`,

Other 3D plotting: `bbox3d()`, `cross3d()`, `ellipse3d.axes()`, `heplot3d()`

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")
```

---

bartlettTests *Bartlett Tests of Homogeneity of Variances*

Description

This function extends `stats::bartlett.test()` to a multivariate response setting. It performs the Bartlett test of homogeneity of variances for each of a set of response variables, and prints a compact summary.

Bartlett's test is the univariate version of Box's M test for equality of covariance matrices. This function provides a univariate follow-up test to Box's M test to give one simple assessment of which response variables contribute to significant differences in variances among groups.

Usage

```
bartlettTests(y, ...)

## Default S3 method:
bartlettTests(y, group, ...)

## S3 method for class 'formula'
bartlettTests(y, data, ...)

## S3 method for class 'lm'
bartlettTests(y, ...)
```

Arguments

y	A data frame or matrix of numeric response variables for the default method, or a model formula for a multivariate linear model, or the multivariate linear model itself. In the case of a formula or model, the variables on the right-hand-side of the model must all be factors and must be completely crossed.
...	other arguments, passed to `stats::bartlett.test()`

group	a vector or factor object giving the group for the corresponding elements of the rows of y for the default method
data	the data set, for the formula method

Value

An object of classes "anova" and "data.frame", with one observation for each response variable in y.

Author(s)

Michael Friendly

References

Bartlett, M. S. (1937). Properties of sufficiency and statistical tests. *Proceedings of the Royal Society of London Series A*, **160**, 268-282.

See Also

boxM() for Box's M test for all responses together.

Other homogeneity tests: leveneTests()

Examples

```
bartlettTests(iris[,1:4], iris$Species)

data(Skulls, package="heplots")
bartlettTests(Skulls[,-1], Skulls$epoch)

# formula method
bartlettTests(cbind(mb, bh, bl, nh) ~ epoch, data=Skulls)
```

---

bbox3d *Find the bounding box of a* rgl::mesh3d *or* rgl::qmesh3d *object*

Description

Ellipsoids are created by rgl functions as meshes of points, segments, ... from coordinates in various forms. This function calculates the bounding box, defined as the range of the x, y, and z coordinates.

Usage

```
bbox3d(x, ...)
```

Arguments

x	A mesh3d object
...	ignored

Value

A 2 x 3 matrix, giving the minimum and maximum values in the rows and x, y, z coordinates in the columns.

See Also

Other 3D plotting: arrow3d(), cross3d(), ellipse3d.axes(), heplot3d()

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.

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

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.

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)
require(car)

# 3-way factorial, ignoring 0 group
bees.mod <- lm(cbind(Iz,Iy) ~ caste*treat*time, data=Bees)
car::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, size="effect",
    xlab="Iz: Ovarian development",
    ylab="Iz: Ovarian reabsorption",
    main="Bees: ~caste*treat*time",
    )

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

# HE plots for this model, with both significance and effect size scaling

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)
}
```

---

boxM *Box's M-test for Homogeneity of Covariance Matrices*

Description

boxM() performs the Box's (1949) M-test for homogeneity of covariance matrices obtained from multivariate normal data according to one or more classification factors. The test compares the product of the log determinants of the separate covariance matrices to the log determinant of the pooled covariance matrix, analogous to a likelihood ratio test. The test statistic uses a chi-square approximation.

Usage

```
boxM(Y, ...)

## S3 method for class 'formula'
boxM(Y, data, ...)

## S3 method for class 'lm'
boxM(Y, ...)

## Default S3 method:
boxM(Y, group, ...)

## S3 method for class 'boxM'
print(x, ...)

## S3 method for class 'boxM'
summary(
```

```
  object,
  digits = getOption("digits") - 2,
  cov = FALSE,
  quiet = FALSE,
  ...
)
```

Arguments

Y	The response variable matrix for the default method, or a "mlm" or "formula" object for a multivariate linear model. If Y is a linear-model object or a formula, the variables on the right-hand-side of the model must all be factors and must be completely crossed, e.g., A:B
...	Other arguments passed down
data	A data frame containing the variables in the model. Used only for the formula method.
group	A vector specifying the groups. Used only for the default method.
x	a class "boxM" object, for the print() method
object	A "boxM" object, result of a call to boxM
digits	Number of digits in printed output
cov	Logical; if TRUE, the covariance matrices for each group and the pooled covariance matrix are printed
quiet	Logical; if TRUE, suppress printed output

Details

As an object of class "boxM", a few methods are available: print.boxM(), summary.boxM() and plot.boxM().

There is no general provision as yet for handling missing data. Missing data are simply removed, with a warning.

As well, the computation assumes that the covariance matrix for each group is non-singular, so that $\log det(S_i)$ can be calculated for each group. At the minimum, this requires that $n > p$ for each group.

Box's M test for a multivariate linear model highly sensitive to departures from multivariate normality, just as the analogous univariate test. It is also affected adversely by unbalanced designs. Some people recommend to ignore the result unless it is very highly significant, e.g., p < .0001 or worse.

In general, heterogeneity of covariance matrices can be more easily seen and understood by plotting the covariance ellipses using covEllipses().

The summary method prints a variety of additional statistics based on the eigenvalues of the covariance matrices. These are returned invisibly, as a list containing the following components:

logDet the vector of log determinants

eigs eigenvalues of the covariance matrices

eigstats statistics computed on the eigenvalues for each covariance matrix:

product the product of eigenvalues, $\prod \lambda_i$
sum the sum of eigenvalues, $\sum \lambda_i$
precision the average precision of eigenvalues, $1/\sum(1/\lambda_i)$
max the maximum eigenvalue, λ_1

Value

A list with class `c("boxM", "htest")` containing the following components:

`statistic`	the chi-square (approximate) statistic for Box's M test, where large values imply the covariance matrices differ.
`parameter`	the degrees of freedom for the test statistic.
`p.value`	the p-value of the test
`ngroups`	the number of levels of the `group` variable
`cov`	a list of the group covariance matrices, of length `ngroups`
`pooled`	the pooled covariance matrix
`means`	a matrix whose `ngroups+1` rows are the means of the variables, followed by those for pooled data.
`logDet`	a vector of length `ngroups+1` containing the natural logarithm of each matrix in `cov`, followed by that for the pooled covariance matrix
`df`	a vector of the degrees of freedom for all groups, followed by that for the pooled covariance matrix
`data.name`	a character string giving the names of the data, as extracted from the call
`method`	the character string `"Box's M-test for Homogeneity of Covariance Matrices"`

Author(s)

The default method was taken from the `biotools` package, Anderson Rodrigo da Silva `<anderson.agro@hotmail.com>`

Generalized by Michael Friendly and John Fox

References

Box, G. E. P. (1949). A general distribution theory for a class of likelihood criteria. *Biometrika*, 36, 317-346.

Morrison, D.F. (1976) *Multivariate Statistical Methods*.

See Also

`car::leveneTest()` carries out homogeneity of variance tests for univariate models with better statistical properties.

`plot.boxM()`, a simple dot plot of the log determinants compared with that of the pooled covariance matrix, and also of other quantities computed from their eigenvalues

`covEllipses()` plots covariance ellipses in variable space for several groups.

Examples

```
data(iris)

# default method, using `Y`, `group`
res <- boxM(iris[, 1:4], iris[, "Species"])
res

# summary method gives details
summary(res)

# visualize (this is what is done in the plot method)
dets <- res$logDet
ng <- length(res$logDet)-1
dotchart(dets, xlab = "log determinant")
points(dets , 1:4, cex=c(rep(1.5, ng), 2.5), pch=c(rep(16, ng), 15),
       col= c(rep("blue", ng), "red"))

# plot method gives confidence intervals for logDet
plot(res, gplabel="Species")

# formula method
boxM( cbind(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) ~ Species,
      data=iris)

### Skulls data
data(Skulls)

# lm method
skulls.mod <- lm(cbind(mb, bh, bl, nh) ~ epoch, data=Skulls)
skulls.boxM <- boxM(skulls.mod) |>
  print()
summary(skulls.boxM)
```

---

coefplot *Coefficient plots for Multivariate Linear Models*

Description

Displays bivariate confidence ellipses for all parameters in an multivariate linear model, for a given pair of variables. In contrast to univariate coefficient plots for an ordinary linear model (e.g., `parameters::model_parameters()`, plotted via its `plot()` method), which show confidence intervals for parameters one at a time, these plots show how each predictor moves a pair of responses jointly, in a way that can readily be compared.

Usage

```
coefplot(object, ...)
```

```
## S3 method for class 'mlm'
coefplot(
  object,
  variables = 1:2,
  parm = NULL,
  df = NULL,
  level = 0.95,
  intercept = FALSE,
  std = FALSE,
  Scheffe = FALSE,
  bars = TRUE,
  fill = FALSE,
  fill.alpha = 0.2,
  labels = !add,
  label.pos = NULL,
  xlab,
  ylab,
  xlim = NULL,
  ylim = NULL,
  axes = TRUE,
  main = "",
  add = FALSE,
  lwd = 1,
  lty = 1,
  pch = 19,
  col = palette(),
  cex = 2,
  cex.label = 1.5,
  cex.lab = par("cex.lab"),
  lty.zero = 3,
  col.zero = 1,
  pch.zero = "+",
  verbose = FALSE,
  ...
)
```

Arguments

object	A multivariate linear model, such as fit by `lm(cbind(y1, y2, ...) ~ terms, ...)`
...	Other parameters passed to `graphics::plot()`
variables	Response variables to plot, given as their indices or names
parm	Parameters to plot, given as their indices or names
df	Degrees of freedom for hypothesis tests
level	Confidence level for the confidence ellipses
intercept	logical. Include the intercept?

std	logical. If TRUE, plot standardized coefficients instead – see stdmodel() for the standardization convention (response(s) and numeric predictors are standardized; factor predictors are left on their raw 0/1 scale). Not generally useful together with intercept = TRUE, since the intercept becomes ~0 once standardized.
Scheffe	If TRUE, confidence intervals for all parameters have Scheffe coverage, otherwise, individual coverage.
bars	Draw univariate confidence intervals for each of the variables?
fill	a logical value or vector. TRUE means the confidence ellipses will be filled.
fill.alpha	Opacity of the confidence ellipses
labels	Labels for the confidence ellipses
label.pos	Positions of the labels for each ellipse. See label.ellipse()
xlab, ylab	x, y axis labels
xlim, ylim	Axis limits
axes	Draw axes?
main	Plot title
add	logical. Add to an existing plot?
lwd	Line widths
lty	Line types
pch	Point symbols for the parameter estimates
col	Colors for the confidence ellipses, points, lines
cex	Character size for points showing parameter estimates
cex.label	Character size for ellipse labels
cex.lab	Character size for axis labels. Defaults to par("cex.lab").
lty.zero, col.zero, pch.zero	Line type, color and point symbol for horizontal and vertical lines at 0, 0. These default to lty.zero = 3, col.zero = 1 (black) and pch.zero = '+'.
verbose	logical. Print parameter estimates and variance-covariance for each parameter?

Details

This function is also a generalization of car::confidenceEllipse() to a multivariate setting. Note that confidenceEllipse() also has an mlm method (via car::confidenceEllipse()), but it answers a different question: it fixes a *pair of coefficients* (for one or more responses) as the plot axes, and shows their joint confidence region. coefplot() instead fixes a *pair of responses* as the axes and overlays one ellipse per predictor – use it when the question is "how does each predictor move these two responses together?", and confidenceEllipse() when the question is about the relationship between two specific coefficients.

Value

Returns invisibly a list of the coordinates of the ellipses drawn

Author(s)

Michael Friendly

See Also

`car::confidenceEllipse()`, `parameters::model_parameters()`

Other multivariate linear models: `glance.mlm()`, `stdcoef()`, `stdmodel()`

Examples

```
rohwer.mlm <- lm(cbind(SAT,PPVT,Raven)~n+s+ns, data=Rohwer)

coefplot(rohwer.mlm, lwd=2,
         main="Bivariate coefficient plot for SAT and PPVT", fill=TRUE)
coefplot(rohwer.mlm, add=TRUE, Scheffe=TRUE, fill=TRUE)

coefplot(rohwer.mlm, var=c(1,3))

mod1 <- lm(cbind(SAT,PPVT,Raven)~n+s+ns+na+ss, data=Rohwer)
coefplot(mod1, lwd=2, fill=TRUE, parm=(1:5),
 main="Bivariate 68% coefficient plot for SAT and PPVT", level=0.68)

# standardized coefficients, with a factor predictor (SES) in the model
# but excluded from the plotted parm range
mod2 <- lm(cbind(SAT,PPVT,Raven) ~ SES+n+s+ns+na+ss, data=Rohwer)
coefplot(mod2, parm=2:6, std=TRUE, fill=TRUE, level=0.68)
```

---

colDevs *Calculate column deviations from central values*

Description

`colDevs` calculates the column deviations of data values from a central value (mean, median, etc.), possibly stratified by a grouping variable.

Usage

```
colDevs(x, group, center = mean, group.var = FALSE, ...)
```

Arguments

`x`	A numeric data frame or matrix.
`group`	A factor (or variable that can be coerced to a factor) indicating the membership of each observation in `x` in one or more groups. If missing, all the data is treated as a single group. You can also specify the `base::interaction()` of two or more factors.

center	A function used to center the values (for each group if group is specified. The function must take a vector argument and return a scalar result.
group.var	logical or character. If TRUE, the group variable containing factor levels is prepended to the matrix of deviations. If a character variable, this is taken as the name to be used for the group variable.
...	Arguments passed down

Details

Conceptually, the function is similar to a column-wise base::sweep(), by group, allowing an arbitrary center function.

Non-numeric columns of x are removed, with a warning.

Value

By default, it returns a numeric matrix containing the deviations from the centering function. If levels==TRUE, it returns a data.frame containing the group factor prepended to the matrix of deviations.

Author(s)

Michael Friendly

See Also

base::colMeans() for column means,

base::sweep()

Examples

```
data(iris)

Species <- iris$Species
irisdev <- colDevs(iris[,1:4], Species, mean)

irisdev <- colDevs(iris[,1:4], Species, median)
# trimmed mean, using an anonymous function
irisdev <- colDevs(iris[,1:4], Species, function(x) mean(x, trim=0.25))

# include the group factor in output
irisdev <- colDevs(iris[,1:4], Species, group.var = "Species")
head(irisdev)

# no grouping variable: deviations from column grand means
# include all variables (but suppress warning for this doc)
irisdev <- suppressWarnings( colDevs(iris) )

# two-way design
colDevs(Plastic[,1:3], Plastic[,"rate"])
colDevs(Plastic[,1:3], Plastic[,"additive"])
```

```
# cell deviations
#' colDevs(Plastic[,1:3], interaction(Plastic[,c("rate", "additive")]))
```

`covEllipses` *Draw classical and Robust Covariance Ellipses for one or more groups*

Description

The function draws covariance ellipses for one or more groups and optionally for the pooled total sample. It uses either the classical product-moment covariance estimate, or a robust alternative, as provided by `MASS::cov.rob()`. Provisions are provided to do this for more than two variables, in a scatterplot matrix format.

These plot methods provide one way to visualize possible heterogeneity of within-group covariance matrices in a one-way MANOVA design. When covariance matrices are nearly equal, their covariance ellipses should all have the same shape. When centered at a common mean, they should also all overlap.

They can also be used to visualize the difference between classical and robust covariance matrices by overlaying the two in a single plot (via `add=TRUE`).

Usage

```
covEllipses(x, ...)

## S3 method for class 'data.frame'
covEllipses(
  x,
  group,
  pooled = TRUE,
  method = c("classical", "mve", "mcd"),
  ...
)

## S3 method for class 'matrix'
covEllipses(
  x,
  group,
  pooled = TRUE,
  method = c("classical", "mve", "mcd"),
  ...
)

## S3 method for class 'formula'
covEllipses(x, data, ...)

## S3 method for class 'boxM'
```

```
covEllipses(x, ...)

## Default S3 method:
covEllipses(
  x,
  means,
  df,
  labels = NULL,
  variables = 1:2,
  level = 0.68,
  segments = 60,
  center = FALSE,
  center.pch = "+",
  center.cex = 2,
 col = getOption("heplot.colors", c("red", "blue", "black", "darkgreen", "darkcyan",
    "brown", "magenta", "darkgray")),
  lty = 1,
  lwd = 2,
  fill = FALSE,
  fill.alpha = 0.3,
  label.pos = 0,
  xlab,
  ylab,
  vlabels,
  var.cex = 2,
  main = "",
  xlim,
  ylim,
  axes = TRUE,
  offset.axes,
  add = FALSE,
  ...
)
```

Arguments

x	The generic argument. For the default method, this is a list of covariance matrices. For the data.frame and matrix methods, this is a numeric matrix of two or more columns supplying the variables to be analyzed.
...	Other arguments passed to the default method for plot, text, and points
group	a factor defining groups, or a vector of length n=nrow(x) doing the same. If missing, a single covariance ellipse is drawn.
pooled	Logical; if TRUE, the pooled covariance matrix for the total sample is also computed and plotted
method	the covariance method to be used: classical product-moment ("classical"), or minimum volume ellipsoid ("mve"), or minimum covariance determinant ("mcd").

data	For the formula method, a data.frame in which to evaluate.
means	For the default method, a matrix of the means for all groups (followed by the grand means, if pooled=TRUE). Rows are the groups, and columns are the variables. It is assumed that the means have column names corresponding to the variables in the covariance matrices.
df	For the default method, a vector of the degrees of freedom for the covariance matrices
labels	Either a character vector of labels for the groups, or TRUE, indicating that group labels are taken as the names of the covariance matrices. Use labels="" to suppress group labels, e.g., when add=TRUE
variables	indices or names of the response variables to be plotted; defaults to 1:2. If more than two variables are supplied, the function plots all pairwise covariance ellipses in a scatterplot matrix format.
level	equivalent coverage of a data ellipse for normally-distributed errors, defaults to 0.68.
segments	number of line segments composing each ellipse; defaults to 40.
center	If TRUE, the covariance ellipses are centered at the centroid.
center.pch	character to use in plotting the centroid of the data; defaults to "+".
center.cex	size of character to use in plotting the centroid (means) of the data; defaults to 2.
col	a color or vector of colors to use in plotting ellipses— recycled as necessary— see Details. A single color can be given, in which case it is used for all ellipses. For convenience, the default colors for all plots produced in a given session can be changed by assigning a color vector via options(heplot.colors =c(...). Otherwise, the default colors are c("red", "blue", "black", "darkgreen", "darkcyan", "magenta", "brown", "darkgray").
lty	vector of line types to use for plotting the ellipses— recycled as necessary— see Details. Defaults to 1.
lwd	vector of line widths to use for plotting the ellipses— recycled as necessary— see Details. Defaults to 2.
fill	A logical vector indicating whether each ellipse should be filled or not— recycled as necessary— see Details. Defaults to FALSE.
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.
label.pos	Label position, a vector of integers (in 0:4) or character strings (in c("center", "bottom", "left", "top", "right")) use in labeling ellipses, recycled as necessary. Values of 1, 2, 3 and 4, respectively indicate positions below, to the left of, above and to the right of the max/min coordinates of the ellipse; the value 0 (the default) specifies the centroid of the ellipse object. Compass directions and fractions are also allowed; see label.ellipse(). label.pos = NULL uses the correlation of each ellipse to determine "top" (r>=0) or "bottom" (r<0).
xlab	x-axis label; defaults to name of the x variable.

ylab	y-axis label; defaults to name of the y variable.
vlabels	Labels for the variables can also be supplied through this argument, which is more convenient when length(variables) > 2.
var.cex	character size for variable labels in the pairs plot, when length(variables) > 2.
main	main plot label; defaults to "", and presently has no effect.
xlim	x-axis limits; if absent, will be computed from the data.
ylim	y-axis limits; if absent, will be computed from the data.
axes	Whether to draw the x, y axes; defaults to TRUE
offset.axes	proportion to extend the axes in each direction if computed from the data; optional.
add	if TRUE, add to the current plot; the default is FALSE. This argument is has no effect when more than two variables are plotted.

Details

The arguments labels, col, lty, lwd, fill, fill.alpha and label.pos are used to draw the ellipses for the groups and also for the pooled, within-group covariance, which is the **last** in a list when these are computed by the functions. These arguments are each taken in the order specified, and recycled as necessary.

Value

Nothing is returned. The function is used for its side-effect of producing a plot. %Returns invisibly an object of class "covEllipse", %which is a list of the coordinates for the ellipses drawn.

Author(s)

Michael Friendly

See Also

heplot(), boxM(),

MASS::cov.rob()

Other covariance ellipses: ellipse.axes(), ellipse.box(), ellipse3d.axes()

Examples

```
data(iris)

# compare classical and robust covariance estimates
covEllipses(iris[,1:4], iris$Species)
# use `method="mve"`
covEllipses(iris[,1:4], iris$Species,
    fill=TRUE, method="mve", add=TRUE, labels="")
```

```
# method for a boxM object
iris.boxM <- boxM(iris[, 1:4], iris[, "Species"])
iris.boxM
# show the associated covariance ellipses
covEllipses(iris.boxM, fill=c(rep(FALSE,3), TRUE) )
# use centering
covEllipses(iris.boxM, fill=c(rep(FALSE,3), TRUE), center=TRUE, label.pos=1:4 )

# method for a list of covariance matrices
cov <- c(iris.boxM$cov, pooled=list(iris.boxM$pooled))
df <- c(table(iris$Species)-1, nrow(iris)-3)
covEllipses(cov, iris.boxM$means, df, label.pos=3, fill=c(rep(FALSE,3), TRUE))

covEllipses(cov, iris.boxM$means, df, label.pos=3, fill=c(rep(FALSE,3), TRUE), center=TRUE)

# scatterplot matrix version, specifying `variables`
covEllipses(iris[,1:4], iris$Species,
 fill=c(rep(FALSE,3), TRUE), variables=1:4,
 fill.alpha=.1)
```

---

cqplot *Chi Square Quantile-Quantile plots*

Description

A chi square quantile-quantile plots show the relationship between data-based values which should be distributed as χ^2 and corresponding quantiles from the χ^2 distribution. In multivariate analyses, this is often used both to assess multivariate normality and check for or identify outliers.

For a data frame of numeric variables or a matrix supplied as the argument `x`, it uses the Mahalanobis squared distances (D^2) of observations $\mathbf{x}$ from the centroid $\bar{\mathbf{x}}$ taking the sample covariance matrix $\mathbf{S}$ into account,

$$D^2 = (\mathbf{x} - \bar{\mathbf{x}})' \, \mathbf{S}^{-1} \, (\mathbf{x} - \bar{\mathbf{x}}) \; .$$

The method for `"mlm"` objects fit using `stats::lm()` for a multivariate response applies this to the residuals from the model.

Usage

```
cqplot(x, ...)

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

## Default S3 method:
cqplot(
  x,
```

```
  method = c("classical", "mcd", "mve"),
  detrend = FALSE,
  pch = 19,
  col = palette()[1],
  cex = par("cex"),
  ref.col = "red",
  ref.lwd = 2,
  conf = 0.95,
  env.col = "gray",
  env.lwd = 2,
  env.lty = 1,
  env.fill = TRUE,
  fill.alpha = 0.2,
  fill.color = trans.colors(ref.col, fill.alpha),
  labels = if (!is.null(rownames(x))) rownames(x) else 1:nrow(x),
  id.n,
  id.method = "r",
  id.cex = 1,
  id.col = palette()[1],
  xlab,
  ylab,
  main,
  what = deparse(substitute(x)),
  ylim,
  ...
)
```

Arguments

x	either a numeric data frame or matrix for the default method, or an object of class "mlm" representing a multivariate linear model. In the latter case, residuals from the model are plotted.
...	Other arguments passed to methods
method	estimation method used for center and covariance, one of: "classical" (product-moment), "mcd" (minimum covariance determinant), or "mve" (minimum volume ellipsoid).
detrend	logical; if FALSE, the plot shows values of D^2 vs. χ^2. if TRUE, the ordinate shows values of $D^2 - \chi^2$
pch	plot symbol for points. Can be a vector of length equal to the number of rows in x.
col	color for points. Can be a vector of length equal to the number of rows in x. The default is the *first* entry in the current color palette (see grDevices::palette() and graphics::par()).
cex	character symbol size for points. Can be a vector of length equal to the number of rows in x.
ref.col	Color for the reference line

ref.lwd	Line width for the reference line
conf	confidence coverage for the approximate confidence envelope
env.col	line color for the boundary of the confidence envelope
env.lwd	line width for the confidence envelope
env.lty	line type for the confidence envelope
env.fill	logical; should the confidence envelope be filled?
fill.alpha	transparency value for fill.color
fill.color	color used to fill the confidence envelope
labels	vector of text strings to be used to identify points, defaults to rownames(x) or observation numbers if rownames(x) is NULL
id.n	number of points labeled. If id.n=0, the default, no point identification occurs.
id.method	point identification method. The default id.method="r" will identify the id.n points with the largest value of abs(y), i.e., the largest Mahalanobis DSQ. See car::showLabels() for other options.
id.cex	size of text for point labels
id.col	color for point labels
xlab	label for horizontal (theoretical quantiles) axis
ylab	label for vertical (empirical quantiles) axis
main	plot title
what	the name of the object plotted; used in the construction of main when that is not specified.
ylim	limits for vertical axis. If not specified, the range of the confidence envelope is used.

Details

cqplot is a more general version of similar functions in other packages that produce chi square QQ plots. It allows for classical Mahalanobis squared distances as well as robust estimates based on the MVE and MCD; it provides an approximate confidence (concentration) envelope around the line of unit slope, a detrended version, where the reference line is horizontal, the ability to identify or label unusual points, and other graphical features.

Cases with any missing values are excluded from the calculation and graph with a warning.

Confidence envelope: In the typical use of QQ plots, it essential to have something in the nature of a confidence band around the points to be able to appreciate whether, and to what degree the observed data points differ from the reference distribution. For cqplot, this helps to assess whether the data are reasonably distributed as multivariate normal and also to flag potential outliers.

The calculation of the confidence envelope here follows that used in the SAS program, http://www.datavis.ca/sasmac/cqplot.html which comes from Chambers et al. (1983), Section 6.8.

The essential formula computes the standard errors as:

$$\text{se}(D^2_{(i)}) = \frac{\hat{b}}{d(q_i)} \times \sqrt{p_i(1-p_i)/n}$$

where $D^2_{(i)}$ is the i-th ordered value of D^2, $\hat{b}$ is an estimate of the slope of the reference line obtained from the ratio of the interquartile range of the D^2 values to that of the χ^2_p distribution and $d(q_i)$ is the density of the chi square distribution at the quantile q_i.

The pointwise confidence envelope of coverage `conf` $= 1 - \alpha$ is then calculated as $D^2_{(i)} \pm z_{1-\alpha/2}\text{se}(D^2_{(i)})$

Note that this confidence envelope applies only to the D^2 computed using the classical estimates of location ($\bar{\mathbf{x}}$) and scatter ($\mathbf{S}$). The `car::qqPlot()` function provides for simulated envelopes, but only for a univariate measure. Oldford (2016) provides a general theory and methods for QQ plots.

Value

Returns invisibly a data.frame containing squared Mahalanobis distances (`DSQ`), their `quantiles` and p-values corresponding to the rows of `x` or the residuals of the model for the identified points, else `NULL` if no points are identified.

Author(s)

Michael Friendly

References

J. Chambers, W. S. Cleveland, B. Kleiner, P. A. Tukey (1983). *Graphical methods for data analysis*, Wadsworth.

R. W. Oldford (2016), "Self calibrating quantile-quantile plots", *The American Statistician*, 70, 74-90.

See Also

`Mahalanobis()` for calculation of Mahalanobis squared distance;

`stats::qqplot()`; `car::qqPlot()` can give a similar result for Mahalanobis squared distances of data or residuals; `qqtest::qqtest()` has many features for all types of QQ plots.

Other diagnostic plots: `distancePlot()`, `eigstatCI()`, `plot.boxM()`

Examples

```
cqplot(iris[, 1:4])

iris.mod <- lm(as.matrix(iris[,1:4]) ~ Species, data=iris)
out <- cqplot(iris.mod, id.n=3)
# show return value
out
```

```
# compare with car::qqPlot
car::qqPlot(Mahalanobis(iris[, 1:4]), dist="chisq", df=4)


# Adopted data
Adopted.mod <- lm(cbind(Age2IQ, Age4IQ, Age8IQ, Age13IQ) ~ AMED + BMIQ,
                  data=Adopted)
cqplot(Adopted.mod, id.n=3)
cqplot(Adopted.mod, id.n=3, method="mve")


# Sake data
Sake.mod <- lm(cbind(taste, smell) ~ ., data=Sake)
cqplot(Sake.mod)
cqplot(Sake.mod, method="mve", id.n=2)

# SocialCog data -- one extreme outlier
data(SocialCog)
SC.mlm <-  lm(cbind(MgeEmotions,ToM, ExtBias, PersBias) ~ Dx,
               data=SocialCog)
cqplot(SC.mlm, id.n=1)

# data frame example: stackloss data
data(stackloss)
cqplot(stackloss[, 1:3], id.n=4)                # very strange
cqplot(stackloss[, 1:3], id.n=4, detrend=TRUE)
cqplot(stackloss[, 1:3], id.n=4, method="mve")
cqplot(stackloss[, 1:3], id.n=4, method="mcd")
```

`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 `rgl::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

`rgl::segments3d()`

Other 3D plotting: `arrow3d()`, `bbox3d()`, `ellipse3d.axes()`, `heplot3d()`

---

df.terms *Find degrees of freedom for model terms*

Description

Find degrees of freedom for model terms

Usage

```
df.terms(model, term, ...)

## Default S3 method:
df.terms(model, term, ...)
```

Arguments

`model`	A model object, such as fit using `stats::lm()`.
`term`	One or more terms from the model
`...`	Other arguments, ignored

Diabetes *Diabetes Dataset*

Description

Reaven and Miller (1979) examined the relationship among blood chemistry measures of glucose tolerance and insulin in 145 nonobese adults. They used the PRIM9 system at the Stanford Linear Accelerator Center to visualize the data in 3D, and discovered a peculiar pattern that looked like a large blob with two wings in different directions.

Format

A data frame with 145 observations on the following 6 variables.

`relwt` relative weight, expressed as the ratio of actual weight to expected weight, given the person's height, a numeric vector

`glufast` fasting plasma glucose level, a numeric vector

`glutest` test plasma glucose level, a measure of glucose intolerance, a numeric vector

`instest` plasma insulin during test, a measure of insulin response to oral glucose, a numeric vector

`sspg` steady state plasma glucose, a measure of insulin resistance, a numeric vector

`group` diagnostic group, a factor with levels `Normal` `Chemical_Diabetic` `Overt_Diabetic`

Details

After further analysis, the subjects were classified as subclinical (chemical) diabetics, overt diabetics and normals. This study was influential in defining the stages of development of Type 2 diabetes. Overt diabetes is the most advanced stage, characterized by elevated fasting blood glucose concentration and classical symptoms. Preceding overt diabetes is the latent or chemical diabetic stage, with no symptoms of diabetes but demonstrable abnormality of oral or intravenous glucose tolerance.

`glutest` was defined as the "area under the plasma glucose curve for the three hour oral glucose tolerance test." Reaven & Miller refer to this variable as "Glucose area".

`instest` was defined as the "area under the plasma insulin curve", and is referred to in the paper as "Insulin area".

This study was influential in defining the stages of development of Type 2 diabetes. Overt diabetes is the most advanced stage, characterized by elevated fasting blood glucose concentration and classical symptoms. Preceding overt diabetes is the latent or chemical diabetic stage, with no symptoms of diabetes but demonstrable abnormality of oral or intravenous glucose tolerance.

Source

Andrews, D. F. & Herzberg, A. M. (1985). *Data: A Collection of Problems from Many Fields for the Student and Research Worker*, Springer-Verlag, Ch. 36.

Friendly, M. (1991). *SAS System for Statistical Graphics*, Cary, NC: SAS Institute.

References

Reaven, G. M. and Miller, R. G. (1979). An attempt to define the nature of chemical diabetes using a multidimensional analysis. *Diabetologia*, 16, 17-24.

Examples

```
data(Diabetes)
col <- c("blue", "red", "darkgreen")[Diabetes$group]
pch <- c(16,15,17)[Diabetes$group]

# a perplexing plot, similar to Fig 2, but with a loess smooth
plot(instest ~ glutest, data=Diabetes, pch=16,
```

```
 cex.lab=1.25,
 xlab="Glucose area (glutest)",
 ylab="Insulin area (instest)")
lines( loess.smooth(Diabetes$glutest, Diabetes$instest), col="blue", lwd=2)

# scatterplot matrix, colored by group
plot(Diabetes[,1:5], col=col, pch=pch)

# covariance ellipses
covEllipses(Diabetes[,2:5], Diabetes$group, fill=TRUE, pooled=FALSE,
 col=col)

covEllipses(Diabetes[,2:5], Diabetes$group, fill=TRUE, pooled=FALSE,
 col=col, variables=1:4)

# Box's M test
diab.boxm <- boxM(Diabetes[,2:5], Diabetes$group)
diab.boxm
plot(diab.boxm)

# heplots
diab.mlm <- lm(cbind(glufast, glutest, instest, sspg) ~ group, data=Diabetes)

heplot(diab.mlm)
pairs(diab.mlm, fill=TRUE, fill.alpha=0.1)
```

`distancePlot` *Distance Plot for Multivariate Model Diagnosis*

Description

This plot, suggested by Rousseeuw & van Zomeren (1991), Rousseeuw et al. (2004) typically plots Mahalanobis distances (D) of the `Y` response variables against the distances of the `X` variables in a multivariate linear model (MLM). When applied to a multivariate linear model itself, it plots the distances of the *residuals* for the `Y` variables against the predictor terms in the *model.matrix* `X`. The motivation behind this plot was to contribute to understanding robust methods for multivariate linear models, such as implemented here in `robmlm()`.

This diagnostic plot combines the information on regression outliers and leverage points, and often more useful than either distance separately, because it shows WHY observations are unusual, either in X or Y space or both. In spirit, it is similar to the ideas behind influence plots for univariate models, such as `car::influencePlot()` and the multivariate analogs provided in the `mvinfluence` package via influencePlot().

Usage

```
distancePlot(X, Y, ...)
```

```
## Default S3 method:
distancePlot(
  X,
  Y,
  method = c("classical", "mcd", "mve"),
  level = 0.975,
  ids = rownames(X),
  pch = c(1, 16),
  col = c("black", "red"),
  label.pos = 2,
  xlab,
  ylab,
  verbose = FALSE,
  ...
)

## S3 method for class 'formula'
distancePlot(X, Y, data, ...)

## S3 method for class 'mlm'
distancePlot(X, ...)
```

Arguments

X	A multivariate linear model fit by stats::lm(), or a numeric data frame giving the predictors in the MLM
Y	A numeric data frame giving the responses in the MLM or the residuals
...	Other arguments passed to methods
method	Estimation method used for center and covariance, one of: "classical" (product-moment), "mcd" (minimum covariance determinant), or "mve" (minimum volume ellipsoid).
level	Lower-tail probability beyond which observations will be labeled.
ids	Labels for observations
pch	A vector of two point symbols, for the regular points and those beyond the cutoffs
col	A vector of two colors, for the regular points and those beyond the cutoffs
label.pos	Position of the label relative to the point; see graphics::text()
xlab	Label stub for horizontal axis
ylab	Label stub for vertical axis
verbose	Logical; if TRUE print the cutoff values to the console
data	For the formula method, the dataset containing the variables

Details

Observations with "large" distances on `X` or `Y` are labeled with their `ids`. The cutoffs are calculated as $\sqrt{\chi^2_{k,\text{level}}}$.

The `method` argument provides a specification the covariance matrix used to calculate the Mahalanobis distances shown in this plot.

Value

`Returns invisibly a data frame containing the distances, `distX`, `distY``

References

Rousseeuw, P. J., & van Zomeren, B. C. (1990). Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, **85**(411), 633-639. doi:10.1080/01621459.1990.10474920.

Rousseeuw P. J. & van Zomeren B. C. (1991). "Robust Distances: Simulation and Cutoff Values." In W Stahel, S Weisberg (eds.), *Directions in Robust Statistics and Diagnostics, Part II*. Springer-Verlag, New York.

Rousseeuw, P. J., Van Driessen, K., Van Aelst, S., & Agullo, J. (2004). Robust multivariate regression. *Technometrics*, **46**(3), 293–305. doi:10.1198/004017004000000329.

See Also

`Mahalanobis()`, `robmlm()`

Other diagnostic plots: `cqplot()`, `eigstatCI()`, `plot.boxM()`

Examples

```
if(require("robustbase")) {
  # Examples from Rousseeuw etal (2004)
  data(pulpfiber, package="robustbase")
  # Figure 1
  distancePlot(pulpfiber[, 1:4], pulpfiber[, 5:8])
  # Figure 3
  pulp.mod <- lm(cbind(Y1, Y2, Y3, Y4) ~ X1 + X2 + X3 + X4, data = pulpfiber)
  distancePlot(pulp.mod, method = "mcd")
}

# NLSY data
data(NLSY, package = "heplots")
NLSY.mlm <- lm(cbind(math, read) ~ income + educ + antisoc + hyperact,
               data = NLSY)

distancePlot(NLSY.mlm)

# gives the same result
distancePlot(NLSY[, 3:6], residuals(NLSY.mlm), level = 0.975)

distancePlot(NLSY.mlm, method ="mve")
```

```
# distancePlot(cbind(math, read) ~ income + educ + antisoc + hyperact,
#                data = NLSY)

# schooldata dataset
data(schooldata)
school.mod <- lm(cbind(reading, mathematics, selfesteem) ~ ., data=schooldata)
distancePlot(school.mod, cex = 1.5, cex.lab = 1.2)

data(Hernior)
Hern.mod <- lm(cbind(leave, nurse, los) ~
               age + sex +  pstat +  build + cardiac + resp, data=Hernior)
distancePlot(Hern.mod)
```

`dogfood` *Dogfood Preferences*

Description

A tiny hypothetical dataset to illustrate one-way MANOVA.

A dogfood manufacturer wanted to study preference for different dogfood formulas, two of their own ("Old", "New") and two from other manufacturers ("Major", "Alps"). In a between-dog design, 4 dogs were presented with a bowl of one `formula` and the time to `start` eating and `amount` eaten were recorded.

Usage

```
data("dogfood")
```

Format

A data frame with 16 observations on the following 3 variables.

`formula` factor, a factor with levels `Old`, `New`, `Major`, `Alps`

`start` numeric, time to start eating

`amount` numeric, amount eaten

Details

In addition to testing the overall effects of `formula`, three useful (and orthogonal) contrasts can specified for this 3-df factor:

- `Ours` vs. `Theirs`, with weights `c(1, 1, -1, -1)`
- `Major` vs. `Alps`, with weights `c(0, 0, 1, -1)`
- `Old` vs. `New`, with weights `c(1, -1, 0, 0)`

Because these are orthogonal contrasts, they fully decompose the main effect of `formula`, in that their sum of squares add to the overall sum of squares.

Source

Used in my Psych 6140 lecture notes, http://friendly.apps01.yorku.ca/psy6140/

Examples

```
data(dogfood)
library(car)
library(candisc)

# make some boxplots
op <- par(mfrow = c(1,2))
boxplot(start ~ formula, data = dogfood)
points(start ~ formula, data = dogfood, pch=16, cex = 1.2)

boxplot(amount ~ formula, data = dogfood)
points(amount ~ formula, data = dogfood, pch=16, cex = 1.2)
par(op)

# setup contrasts to test interesting comparisons
C <- matrix(
       c( 1,  1, -1, -1,         #Ours vs. Theirs
          0,  0,  1, -1,           #Major vs. Alps
          1, -1,  0,  0),             #New vs. Old
       nrow=4, ncol=3)
# assign these to the formula factor
contrasts(dogfood$formula) <- C
# re-fit the model
dogfood.mod <- lm(cbind(start, amount) ~ formula, data=dogfood)

dogfood.mod <- lm(cbind(start, amount) ~ formula, data=dogfood)
Anova(dogfood.mod)

# data ellipses
covEllipses(cbind(start, amount) ~ formula, data=dogfood,
  fill = TRUE, fill.alpha = 0.1)

# test these contrasts with multivariate tests
linearHypothesis(dogfood.mod, "formula1", title="Ours vs. Theirs")
linearHypothesis(dogfood.mod, "formula2", title="Old vs. New")
linearHypothesis(dogfood.mod, "formula3", title="Alps vs. Major")

heplot(dogfood.mod, fill = TRUE, fill.alpha = 0.1)

# display contrasts in the heplot
hyp <- list("Ours/Theirs" = "formula1",
            "Old/New" = "formula2")
heplot(dogfood.mod, hypotheses = hyp,
       fill = TRUE, fill.alpha = 0.1)

dogfood.can <- candisc(dogfood.mod, data=dogfood)
heplot(dogfood.can,
       fill = TRUE, fill.alpha = 0.1,
```

```
          lwd = 2, var.lwd = 2, var.cex = 2)
```

`eigstatCI` *Bootstrap Confidence Intervals for Eigenvalue Statistics*

Description

Computes bootstrap confidence intervals for statistics based on eigenvalues of grouped covariance matrices. This is intended for use with `plot.boxM()` to provide confidence intervals for measures beyond the default "logDet".

This is a convenience wrapper that extracts the necessary information from a boxM object, but still requires the original data and grouping variable since these are not stored in the boxM object.

Usage

```
eigstatCI(
  Y,
  group,
  which = c("product", "sum", "precision", "max"),
  R = 1000,
  conf = 0.95,
  type = c("perc", "bca", "norm", "basic"),
  parallel = FALSE,
  ncpus = 2,
  seed = NULL,
  ...
)

eigstatCI_boxM(boxm, Y, group, ...)
```

Arguments

`Y`	Original data matrix used in `boxM()`
`group`	Original grouping variable used in `boxM()`
`which`	The eigenvalue-based statistic to compute. One of: **product** Product of eigenvalues (= determinant) **sum** Sum of eigenvalues (= trace) **precision** Harmonic mean of eigenvalues **max** Maximum eigenvalue
`R`	Number of bootstrap replicates. Default is 1000.
`conf`	Confidence level for intervals ($0 < \text{conf} < 1$). Default is 0.95.
`type`	Type of bootstrap confidence interval. Options: **perc** Percentile method (default, most robust) **bca** Bias-corrected and accelerated

norm Normal approximation

basic Basic bootstrap

`parallel`	Logical or character string. If TRUE, use parallel processing via the boot package. Can also be "multicore" or "snow". Default is FALSE.
`ncpus`	Number of CPUs to use if parallel=TRUE. Default is 2.
`seed`	Random seed for reproducibility. If NULL, no seed is set.
`...`	Additional arguments passed to `eigstatCI()`
`boxm`	A "boxM" object from `boxM()`

Details

For each group (and the pooled data), this function performs nonparametric bootstrap resampling to estimate the sampling distribution of the specified eigenvalue-based statistic. The "pooled" statistic is computed from the pooled within-group covariance matrix (as in `boxM()$pooled`), and is bootstrapped by resampling within groups (a stratified bootstrap). Confidence intervals are computed using the percentile method or bias-corrected and accelerated (BCa) method.

Unlike `logdetCI()` which uses analytic approximations based on asymptotic theory, this function makes no distributional assumptions and can handle small to moderate sample sizes, though computational cost increases with the number of bootstrap replicates.

Value

A data frame with one row for each group plus the pooled data. Columns include:

group Group name (factor level)

statistic Observed value of the statistic

lower Lower confidence limit

upper Upper confidence limit

bias Bootstrap estimate of bias (if available)

se Bootstrap standard error (if available)

A data frame with bootstrap confidence intervals

Author(s)

Michael Friendly

References

Efron, B., & Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. CRC Press.

See Also

`boxM()`, `plot.boxM()`, `logdetCI()`

Other diagnostic plots: `cqplot()`, `distancePlot()`, `plot.boxM()`

Examples

```
## Not run:
library(boot)
data(iris)

# Bootstrap CI for product of eigenvalues
CI_prod <- eigstatCI(iris[,1:4], iris$Species, which="product", R=500)
CI_prod

# Bootstrap CI for sum of eigenvalues (= trace)
CI_sum <- eigstatCI(iris[,1:4], iris$Species, which="sum", R=500)
CI_sum

# Use with parallel processing for speed
CI_max <- eigstatCI(iris[,1:4], iris$Species, which="max",
                    R=1000, parallel=TRUE, ncpus=4)

## End(Not run)

## Not run:
library(boot)
data(iris)

# Fit boxM
res <- boxM(iris[,1:4], iris$Species)

# Get bootstrap CIs (must provide original data again)
CI <- eigstatCI_boxM(res, Y = iris[,1:4], group = iris$Species,
                     which = "sum", R = 500)

## End(Not run)
```

`ellipse.axes` *Draw Axes of a 2D Covariance Ellipse*

Description

A function to draw the principal axes of a 2D ellipse from a correlation, covariance or sums of squares and cross products matrix in an existing plot.

Usage

```
ellipse.axes(
  x,
  centre = c(0, 0),
  center = centre,
  scale,
  which = 1:2,
```

```
  level = 0.95,
  radius = sqrt(qchisq(level, 2)),
  extend = 0,
  labels = TRUE,
  label.ends = c(2, 4),
  label.pos = c(2, 4, 1, 3),
  type = c("lines", "arrows"),
  ...
)
```

Arguments

x	A square positive definite matrix at least 2×2 in size. It will be treated as the correlation or covariance of a multivariate normal distribution.
centre, center	The center of the ellipse
scale	If x is a correlation matrix, then the standard deviations of each parameter can be given in the scale parameter. This defaults to `c(1, 1)`, so no rescaling will be done.
which	An integer vector to select which variables from the object `x` will be plotted. The default is the first 2.
level	The coverage level of a simultaneous region of the ellipse. The default is 0.95, for a 95% region. This is used to control the size of the ellipse.
radius	The size of the ellipsoid may also be controlled by specifying the value of a t-statistic on its boundary. This defaults to the square root of a chi-square statistic for a given `level` on 2 degrees of freedom, however in a small sample of `n` observations, a more accurate value is `sqrt(2 * qf(level, 2, n - 1 ))`.
extend	Fraction to extend the `radius` (default: 0). For example, use `extend = 0.1` to extend the ellipse axes by 10%.
labels	Either a logical value, a character string, or a character vector of length 2. If `TRUE`, the default, the axes are labeled "PC1", "PC2". If a single character string, the digits 1, and 2 are pasted on the end.
label.ends	A vector of indices in the range `1:4` indicating which ends of the axes should be labeled, corresponding to a selection of rows of the 4 x 2 matrix of axes end points. Values `1:2` represent the minimum and maximum of the first dimension respectively. Values `3:4` represent the minimum and maximum of the second dimension. Default: `c(2, 4)`.
label.pos	Positions of text labels relative to the ends of the axes used in `graphics::text()` for the four possible `label.ends`. 1, 2, 3, 4 represent below, to the left, above and to the right. The default, `c(2, 4, 1, 3)`, positions the labels outside the axes.
type	Character. Draw `"lines"` or `"arrows"`.
...	Other arguments passed to `graphics::lines()` and `graphics::text()`.

Value

Invisibly returns a 4 x 2 matrix containing the end points of the axes in pairs (min, max) by rows.

Author(s)

Michael Friendly

See Also

graphics::lines(), graphics::text()

Other covariance ellipses: covEllipses(), ellipse.box(), ellipse3d.axes()

Examples

```
data(iris)
cov <- cov(iris[,1:2])
mu <- colMeans(iris[,1:2])

radius <- sqrt(qchisq(0.68, 2))
plot(iris[,1:2], asp=1)
car::ellipse(mu, cov, radius = radius)
res <- ellipse.axes(cov, center=mu, level = 0.68,
                    labels = TRUE)
res

# try some options
plot(iris[,1:2], asp=1)
car::ellipse(mu, cov, radius = radius)
abline(h=mu[2], v=mu[1], col = "grey")
ellipse.axes(cov, centre=mu, level = 0.68,
             labels = "Dim", label.ends = 1:4,
             lwd = 2, lty = 2, col = "red",
             cex = 1.5)

# draw arrows rather than lines
plot(iris[,1:2], asp=1)
car::ellipse(mu, cov, radius = radius)
ellipse.axes(cov, center=mu, level = 0.68,
             type = "arrows", extend = 0.2)
```

---

ellipse.box *Draw Conjugate Axes and Parallelogram Surrounding a Covariance Ellipse*

Description

Draw Conjugate Axes and Parallelogram Surrounding a Covariance Ellipse

Usage

```
ellipse.box(
  x,
  center = c(0, 0),
  which = 1:2,
  level = 0.95,
  radius = sqrt(qchisq(level, 2)),
  factor = c("cholesky", "pca"),
  draw = c("box", "diameters", "both"),
  ...
)
```

Arguments

x	A square positive definite matrix at least 2x2 in size. It will be treated as the correlation or covariance of a multivariate normal distribution.
center	The center of the ellipse
which	An integer vector to select which variables from the object x will be plotted. The default is the first 2.
level	The coverage level of a simultaneous region of the ellipse. The default is 0.95, for a 95\ ellipse.
radius	The size of the ellipsoid may also be controlled by specifying the value of a t-statistic on its boundary. This defaults to the square root of a chi-square statistic for a given level on 2 degrees of freedom, however in a small sample of n observations, a more accurate value is sqrt(2 * qf(level, 2, n - 1)).
factor	A function defining the conjugate axes used to transform the unit circle into an ellipse. base::chol(), uses the right Cholesky factor of x.
draw	What to draw? "box", "diameters" or "both"
...	Other arguments passed to graphics::lines().

Value

Invisibly returns a 2 column matrix containing the end points of lines.

See Also

Other covariance ellipses: covEllipses(), ellipse.axes(), ellipse3d.axes()

Examples

```
data(iris)
cov <- cov(iris[,3:4])
mu <- colMeans(iris[,3:4])

radius <- sqrt(qchisq(0.68, 2))
plot(iris[,3:4], asp=1)
car::ellipse(mu, cov, radius = radius)
```

```
ellipse.axes(cov, center=mu, level = 0.68,
            labels = TRUE)
ellipse.box(cov, center=mu, level = 0.68,
            factor = "pca",
            col = "red", lwd = 2 )

res <- ellipse.box(cov, center=mu, level = 0.68, factor = "chol", col = "green", lwd = 2 )
res
```

`ellipse3d.axes` *Draw axes of a 3D ellipsoid*

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),
  center = centre,
  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

`x`	A square positive definite matrix at least 3x3 in size. It will be treated as the correlation or covariance of a multivariate normal distribution.
`centre, center`	The center of the ellipse
`scale`	If x is a correlation matrix, then the standard deviations of each parameter can be given in the scale parameter. This defaults to `c(1, 1, 1)`, so no rescaling will be done.
`level`	The coverage level of a simultaneous region. The default is 0.95, for a 95\ ellipsoid.
`t`	The size of the ellipsoid may also be controlled by specifying the value of a t-statistic on its boundary, which defaults to the square root of a chi-square statistic for a given `level` on 3 degrees of freedom.
`which`	An integer vector to select which variables from the object will be plotted. The default is the first 3.

`labels`	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.
`label.ends`	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: `c(2,4,6)`.
`...`	Other arguments passed to `segments3d` and `text3d`.

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

`rgl::segments3d()`, `rgl::text3d()`, `rgl::ellipse3d()`

Other covariance ellipses: `covEllipses()`, `ellipse.axes()`, `ellipse.box()`

Other 3D plotting: `arrow3d()`, `bbox3d()`, `cross3d()`, `heplot3d()`

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)
```

`Ellipsoid` *Draw an Ellipsoid in an rgl Scene*

Description

This is an experimental function designed to separate internal code in `heplot3d()`.

Usage

```
Ellipsoid(x, ...)

## S3 method for class 'data.frame'
Ellipsoid(x, which = 1:3, method = c("classical", "mve", "mcd"), ...)

## Default S3 method:
Ellipsoid(
  x,
  center = c(0, 0, 0),
  which = 1:3,
  radius = 1,
  df = Inf,
  label = "",
  cex.label = 1.5,
  col = "pink",
  lwd = 1,
  segments = 40,
  shade = TRUE,
  alpha = 0.1,
  wire = TRUE,
  verbose = FALSE,
  warn.rank = FALSE,
  ...
)
```

Arguments

x	An object. In the default method the parameter x should be a square positive definite matrix at least 3x3 in size. It will be treated as the correlation or covariance of a multivariate normal distribution. For the `data.frame` method, it should be a numeric data frame with at least 3 columns.
...	Other arguments
which	This parameter selects which variables from the object will be plotted. The default is the first 3.
method	the covariance method to be used: classical product-moment (`"classical"`), or minimum volume ellipsoid (`"mve"`), or minimum covariance determinant (`"mcd"`
center	center of the ellipsoid, a vector of length 3, typically the mean vector of data
radius	size of the ellipsoid
df	degrees of freedom associated with the covariance matrix, used to calculate the appropriate F statistic
label	label for the ellipsoid
cex.label	text size of label
col	color of the ellipsoid

lwd	line with for the wire-frame version
segments	number of segments composing each ellipsoid; defaults to 40.
shade	logical; should the ellipsoid be smoothly shaded?
alpha	transparency of the shaded ellipsoid
wire	logical; should the ellipsoid be drawn as a wire frame?
verbose	logical; for debugging
warn.rank	logical; warn if the ellipsoid is less than rank 3?

Value

returns the bounding box of the ellipsoid invisibly; otherwise used for it's side effect of drawing the ellipsoid

Examples

```
# none yet
```

---

etasq *Measures of Partial Association (η^2) 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. For a multivariate model, this is a summary across all response variables.

Usage

```
etasq(x, ...)

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

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

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

Arguments

x	A lm, mlm or Anova.mlm object
...	Other arguments passed down to car::Anova().
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 eta^2.

Details

There is a different analog of η^2 for each of the four standard multivariate test statistics: Pillai's trace, Hotelling-Lawley trace, Wilks' Lambda and Roy's maximum root test.

For univariate linear models, classical η^2 = SSH / SST and partial η^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 $\mathbf{HE}^{-1}$. 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

`car::Anova()`

`effectsize::eta_squared()` for a function that calculates this effect size measure for each response variable separately.

Examples

```
library(car)
data(Soils, package="carData")
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.

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)
require(car)

# 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
if(requireNamespace("rgl")){
heplot3d(foot.mod, variables=c(1,2,6),
 hypotheses=list("group.1"="group1","group.2"="group2"),
 col=c("red", "blue", "green3", "green3" ), wire=FALSE)
}
```

`glance.mlm` *Glance at an mlm object*

Description

This function takes an "mlm" object, fit by `stats::lm()` with a multivariate response. The goal is to return something analogous to `broom::glance.lm()` for a univariate response linear model.

Usage

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

Arguments

`x`	An `"mlm"` object created by `stats::lm()`, i.e., with a multivariate response.
`...`	Additional arguments. Not used.

Details

In the multivariate case, it returns a `tibble::tibble()` with one row for each response variable, containing goodness of fit measures, F-tests and p-values.

Value

A tibble::tibble() with one row for each response variable and the columns:

`r.squared` R squared statistic, or the percent of variation explained by the model.

`sigma` Estimated standard error of the residuals

`fstatitic` Overall F statistic for the model

`numdf` Numerator degrees of freedom for the overall test

`dendf` Denominator degrees of freedom for the overall test

`p.value` P-value corresponding to the F statistic

`nobs` Number of observations used

See Also

Other multivariate linear models: `coefplot()`, `stdcoef()`, `stdmodel()`

Examples

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

---

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

`y`	A numeric data frame or matrix
`order`	An integer vector specifying the order of and/or a subset of the columns of `y` to be orthogonalized. If missing, `order=1:p` where `p=ncol(y)`.
`recenter`	If `TRUE`, the result has same column means as original; else means = 0 for cols `2:p`.
`rescale`	If `TRUE`, the result has same column standard deviations as original; else sd = residual variance for cols `2:p`
`adjnames`	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

In statistical applications, interpretation depends on the `order` of the variables orthogonalized. In multivariate linear models, orthogonalizing the response, Y variables provides the equivalent of step-down tests, where Y1 is tested alone, and then Y2.1, Y3.12, etc. can be tested to determine their additional contributions over the previous response variables.

Similarly, orthogonalizing the model X variables provides the equivalent of Type I tests, such as provided by `stats::anova()`.

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

`base::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)
car::Anova(iris.mod1)

# orthogonalize X side
rohwer.mod <- lm(cbind(SAT, PPVT, Raven) ~ n + s + ns + na + ss, data=Rohwer)
car::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)
car::Anova(rohwer.mod1)
```

```
# compare with anova()
anova(rohwer.mod1)

# compare heplots for original Xs and orthogonalized, Type I
heplot(rohwer.mod)
heplot(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.

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

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).

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

```
library(car)
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",
  label.pos = NULL,
  label.cex = par("cex"),
  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(iterm),
  manova,
  size = c("evidence", "effect.size", "significance"),
  level = 0.68,
  alpha = 0.05,
  segments = 60,
  center.pch = "+",
  center.cex = 2,
 col = getOption("heplot.colors", c("red", "blue", "black", "darkgreen", "darkcyan",
    "magenta", "brown", "darkgray")),
  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".
...	arguments to pass down to plot, text, and points.
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 car::linearHypothesis() function in the car package; the list elements can be named, in which case the names are used.
term.labels	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.
hyp.labels	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.
err.label	Label for the error ellipse
label.pos	Label position, a vector of integers (in 0:4) or character strings (in c("center", "bottom", "left", "top", "right"), or in c("C", "S", "W", "N", "E") use in labeling ellipses, recycled as necessary. Values of 1, 2, 3 and 4, respectively indicate positions below, to the left of, above and to the right of the max/min coordinates of the ellipse; the value 0 specifies the centroid of the ellipse object. The default, label.pos=NULL uses the correlation of the ellipse to determine "top" (r>=0) or "bottom" (r<0). Even more flexible options are described in label.ellipse()
label.cex	Character size used for labels for the hypothesis, error ellipses.
variables	indices or names of the two response variables to be plotted; defaults to 1:2.
error.ellipse	if TRUE, plot the error ellipse; defaults to TRUE, if the argument add is FALSE (see below).
factor.means	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 add is FALSE (see below).
grand.mean	if TRUE, plot the centroid for all of the data; defaults to TRUE, if the argument add is FALSE (see below).

`remove.intercept`
: if `TRUE` (the default), do not plot the ellipse for the intercept even if it is in the MANOVA table.

`type`
: "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).

`idata`
: an optional data frame giving a factor or factors defining the intra-subject model for multivariate repeated-measures data. See Friendly (2010) and Details of `car::Anova()` for an explanation of the intra-subject design and for further explanation of the other arguments relating to intra-subject factors.

`idesign`
: a one-sided model formula using the "data" in idata and specifying the intra-subject design for repeated measure models.

`icontrasts`
: 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 c("contr.sum", "contr.poly").

`imatrix`
: In lieu of `idata` and `idesign`, you can specify the intra-subject design matrix directly via `imatrix`, 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 *different* terms must be mutually orthogonal.

`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 `iterm` effect as well as all interactions of `iterm` with `terms`.

`markH0`
: A logical value (or else a list of arguments to `mark.H0()`) used to draw crosshairs and a point indicating the value of a point null hypothesis. The default is TRUE if `iterm` is non-NULL.

`manova`
: optional `Anova.mlm` object for the model; if absent a MANOVA is computed. Specifying the argument can therefore save computation in repeated calls.

`size`
: 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 `alpha` extends outside of the error ellipse. 'size = "significance"' is a synonym and does the same thing. If '"effect.size"', the hypothesis ellipse is on the same scale as the error ellipse.

`level`
: equivalent coverage of ellipse (assuming normally-distributed errors). This defaults to `0.68`, giving a standard 1 SD bivariate ellipse.

`alpha`
: significance level for Roy's greatest-root test statistic; if `size="evidence"` or `size="significance"`, then the hypothesis ellipse is scaled so that it just touches the error ellipse at the specified alpha level. A larger hypothesis ellipse *somewhere* in the space of all the response variables therefore indicates statistical significance. In a plot of only two responses, an H ellipse inside E does not mean "not significant" (see Details). Defaults to `0.05`.

`segments`
: number of line segments composing each ellipse; defaults to `60`.

`center.pch`
: character to use in plotting the centroid of the data; defaults to `"+"`.

`center.cex`
: size of character to use in plotting the centroid of the data; defaults to `2`.

col	a color or vector of colors to use in plotting ellipses; the first color is used for the error ellipse; the remaining colors — recycled as necessary — are used for the hypothesis ellipses. A single color can be given, in which case it is used for all ellipses. For convenience, the default colors for all heplots produced in a given session can be changed by assigning a color vector via `options(heplot.colors =c(...)`. Otherwise, the default colors are `c("red", "blue", "black", "darkgreen", "darkcyan", "magenta", "brown", "darkgray")`.
lty	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`.
lwd	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`.
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 below.
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.
xlab	x-axis label; defaults to name of the x variable.
ylab	y-axis label; defaults to name of the y variable.
main	main plot label; defaults to `""`.
xlim	x-axis limits; if absent, will be computed from the data.
ylim	y-axis limits; if absent, will be computed from the data.
axes	Whether to draw the x, y axes; defaults to `TRUE`
offset.axes	proportion to extend the axes in each direction if computed from the data; optional.
add	if `TRUE`, add to the current plot; the default is `FALSE`. If `TRUE`, the error ellipse is not plotted.
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 ellipse collapses to a line; the default is `FALSE`.

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`.

With `size="evidence"`, a term is significant by Roy's test if and only if its **H** ellipsoid protrudes outside the **E** ellipsoid *somewhere in the full space of all p responses*. A 2D HE plot shows only the projection onto two responses, so in a plot the rule works in one direction only:

- If **H** extends outside **E** in any view, the term is significant.
- If **H** lies inside **E** in a given view, the term may still be significant, because the effect may lie in directions that view does not show. Check other views with `pairs.mlm()` or `heplot3d()`, a canonical view (`candisc::heplot.candisc()`), or the tests from `car::Anova()`.

For example, in the `Plastic` data, `additive` is significant (Roy's p = .025), but its **H** line stays inside **E** in all three pairwise plots of the responses.

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 $\mathbf{HE}^{-1}$) 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 `car::linearHypothesis()` 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::candisc()` package provides functions for visualizing a multivariate linear model in a low-dimensional view via a generalized canonical discriminant analyses. `candisc::heplot.candisc()` and `candisc::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. % https://www.jstatsoft.org/v17/i06/, DOI: 10.18637/jss.v017.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. DOI: 10.18637/jss.v037.i04.

Fox, J., Friendly, M. & Weisberg, S. (2013). Hypothesis Tests for Multivariate Linear Models Using the car Package. *The R Journal*, **5**(1), https://journal.r-project.org/articles/RJ-2013-004/RJ-2013-004.pdf.

Friendly, M. & Sigal, M. (2014) Recent Advances in Visualizing Multivariate Linear Models. *Revista Colombiana de Estadistica*, **37**, 261-283. DOI: 10.15446/rce.v37n2spe.47934.

See Also

`car::Anova()`, `car::linearHypothesis()` for details on testing MLMs.

`heplot1d()`, `heplot3d()`, `pairs.mlm()`, `mark.H0()` for other HE plot functions. `coefplot.mlm()` for plotting confidence ellipses for parameters in MLMs.

`trans.colors()` for calculation of transparent colors. `label.ellipse()` for labeling positions in plotting **H** and **E** ellipses.

`candisc::candisc()`, `candisc::heplot.candisc()` for reduced-rank views of `mlm`s in canonical space.

Other HE plot functions: `heplot1d()`, `heplot3d()`, `pairs.mlm()`

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; include contrasts
heplot(iris.mod, hypotheses=hyp, fill=TRUE,
       fill.alpha=0.2, col=c("red", "blue"))
heplot(iris.mod, hypotheses=hyp, fill=TRUE,
       col=c("red", "blue"), lty=c(0,0,1,1))

# vary label position and fill.alpha
heplot(iris.mod, hypotheses=hyp, fill=TRUE, fill.alpha=c(0.3,0.1), col=c("red", "blue"),
       lty=c(0,0,1,1), label.pos=0:3)
```

```
# what is returned?
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, package = "carData")
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)
car::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

if(requireNamespace("rgl")){
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.

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", "significance"),
  level = 0.68,
  alpha = 0.05,
  center.pch = "|",
 col = getOption("heplot.colors", c("red", "blue", "black", "darkgreen", "darkcyan",
    "magenta", "brown", "darkgray")),
  lty = 2:1,
  lwd = 1:2,
  xlab,
  main = "",
  xlim,
  axes = TRUE,
  offset.axes = 0.1,
  add = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

mod	a model object of class "mlm".
...	arguments to pass down to plot, text, and points.
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 car::linearHypothesis() function in the car package; the list elements can be named, in which case the names are used.
term.labels	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.

`hyp.labels` 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.

`variables` indices or names of the two response variables to be plotted; defaults to `1:2`.

`error.ellipse` if `TRUE`, plot the error ellipse; defaults to `TRUE`, if the argument `add` is `FALSE` (see below).

`factor.means` 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 `add` is `FALSE` (see below).

`grand.mean` if `TRUE`, plot the centroid for all of the data; defaults to `TRUE`, if the argument `add` is `FALSE` (see below).

`remove.intercept` if `TRUE` (the default), do not plot the ellipse for the intercept even if it is in the MANOVA table.

`type` "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).

`idata` an optional data frame giving a factor or factors defining the intra-subject model for multivariate repeated-measures data. See Details of `car::Anova()` for an explanation of the intra-subject design and for further explanation of the other arguments relating to intra-subject factors.

`idesign` a one-sided model formula using the "data" in idata and specifying the intra-subject design for repeated measure models.

`icontrasts` 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 c("contr.sum", "contr.poly").

`imatrix` In lieu of `idata` and `idesign`, you can specify the intra-subject design matrix directly via `imatrix`, 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 *different* terms must be mutually orthogonal.

`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 `iterm` effect as well as all interactions of `iterm` with `terms`.

`manova` optional `Anova.mlm` object for the model; if absent a MANOVA is computed. Specifying the argument can therefore save computation in repeated calls.

`size` 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 `alpha` extends outside of the error ellipse. 'size = "significance"' is a synonym and does the same thing. If '"effect.size"', the hypothesis ellipse is on the same scale as the error ellipse.

`level` equivalent coverage of ellipse (assuming normally-distributed errors). This defaults to `0.68`, giving a standard 1 SD bivariate ellipse.

alpha	significance level for the univariate F test of the response shown; if size="evidence" or size="significance", then the hypothesis interval is scaled so that it just matches the error interval at the specified alpha level. A longer hypothesis interval therefore indicates that the term is significant for *this response* by its univariate F test. This is not Roy's multivariate test as in heplot(): a term can be significant multivariately but not for any single response. Defaults to 0.05.
center.pch	character to use in plotting the centroid of the data; defaults to "|".
col	a color or vector of colors to use in plotting ellipses; the first color is used for the error ellipse; the remaining colors — recycled as necessary — are used for the hypothesis ellipses. A single color can be given, in which case it is used for all ellipses. For convenience, the default colors for all heplots produced in a given session can be changed by assigning a color vector via options(heplot.colors =c(...). Otherwise, the default colors are c("red", "blue", "black", "darkgreen", "darkcyan", "magenta", "brown", "darkgray").
lty	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.
lwd	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.
xlab	x-axis label; defaults to name of the x variable.
main	main plot label; defaults to "".
xlim	x-axis limits; if absent, will be computed from the data.
axes	Whether to draw the x, y axes; defaults to TRUE
offset.axes	proportion to extend the axes in each direction if computed from the data; optional.
add	if TRUE, add to the current plot; the default is FALSE. If TRUE, the error ellipse is not plotted.
verbose	if TRUE, print the MANOVA table and details of hypothesis tests; the default is FALSE.

Details

In particular, for a given response, the 1-D representations of $\mathbf{H}$ and $\mathbf{E}$ matrices correspond to line segments. The $\mathbf{E}$ ellipse'' is shown as a filled rectangle whose width equals the mean squared error for tha
lipse" 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.

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:

H	ranges for the hypothesis terms
E	range for E
xlim	x-axis limits

Author(s)

Michael Friendly

See Also

car::Anova(), car::linearHypothesis() for hypothesis tests in mlms

heplot(), heplot3d(), pairs.mlm() for other HE plot methods

Other HE plot functions: heplot(), heplot3d(), pairs.mlm()

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 (**H**) and error (**E**) sums-of-squares-and-products matrices for terms and linear hypotheses in a multivariate linear model. It allow you to visualize model effects on three response variables together, as opposed to the 2D views offered by heplot() and pairs.mlm().

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", "significance"),
  level = 0.68,
  alpha = 0.05,
  segments = 40,
 col = getOption("heplot3d.colors", c("red", "blue", "black", "darkgreen", "darkcyan",
    "magenta", "brown", "darkgray")),
  lwd = c(1, 4),
  shade = TRUE,
  shade.alpha = 0.2,
  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,
  cex.label = 1.5,
  add = FALSE,
  verbose = FALSE,
  warn.rank = FALSE,
  ...
)
```

Arguments

mod	a model object of class "mlm".
...	arguments passed from generic.
terms	a logical value or character vector of terms in the model for which to plot hy-

pothesis 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 `car::linearHypothesis()` function in the car package; the list elements can be named, in which case the names are used.

`term.labels` 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.

`hyp.labels` 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.

`err.label` Label for the error ellipse

`variables` indices or names of the three response variables to be plotted; defaults to 1:3.

`error.ellipsoid`
if TRUE, plot the error ellipsoid; defaults to TRUE, if the argument add is FALSE (see below).

`factor.means` 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 add is FALSE (see below).

`grand.mean` if TRUE, plot the centroid for all of the data; defaults to TRUE, if the argument add is FALSE (see below).

`remove.intercept`
if TRUE (the default), do not plot the ellipsoid for the intercept even if it is in the MANOVA table.

`type` "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).

`idata` an optional data frame giving a factor or factors defining the intra-subject model for multivariate repeated-measures data. See Details of `car::Anova()` for an explanation of the intra-subject design and for further explanation of the other arguments relating to intra-subject factors.

`idesign` a one-sided model formula using the "data" in idata and specifying the intra-subject design for repeated measure models.

`icontrasts` 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 c("contr.sum", "contr.poly").

`imatrix` In lieu of `idata` and `idesign`, you can specify the intra-subject design matrix directly via `imatrix`, 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 *different* terms must be mutually orthogonal.

`iterm` For repeated measures designs, you must specify one intra-subject term (a character string) to select the SSPE ($\mathbf{E}$) matrix used in the HE plot. Hypothesis terms plotted include the `iterm` effect as well as all interactions of `iterm` with `terms`.

manova	optional Anova.mlm object for the model; if absent a MANOVA is computed. Specifying the argument can therefore save computation in repeated calls.
size	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 alpha extends outside of the error ellipse. 'size = "significance"' is a synonym and does the same thing. If '"effect.size"', the hypothesis ellipse is on the same scale as the error ellipse.
level	equivalent coverage of ellipse (assuming normally-distributed errors). This defaults to 0.68, giving a standard 1 SD bivariate ellipse.
alpha	significance level for Roy's greatest-root test statistic; if size="evidence" or size="significance", then the hypothesis ellipse is scaled so that it just touches the error ellipse at the specified alpha level. A larger hypothesis ellipse *somewhere* in the space of all the response variables therefore indicates statistical significance. With more than three responses, an H ellipsoid inside E in this 3D view does not mean "not significant": the effect may lie in directions not shown (see Details in heplot()). Defaults to 0.05.
segments	number of segments composing each ellipsoid; defaults to 40.
col	a color or vector of colors to use in plotting ellipsoids; the first color is used for the error ellipsoid; the remaining colors — recycled as necessary — are used for the hypothesis ellipsoid. A single color can be given, in which case it is used for all ellipsoid. For convenience, the default colors for all heplots produced in a given session can be changed by assigning a color vector via options(heplot3d.colors=c(...)). Otherwise, the default colors are c("pink", "blue", "black", "darkgreen", "darkcyan", "magenta", "brown", "darkgray").
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 rgl::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 rgl::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 rgl::bg3d().
fov	field of view angle; controls perspective. See rgl::view3d().
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.
cex.label	text size for ellipse labels
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.

Details

It uses the rgl package for rendering, so it adds some arguments (e.g., fogtype) to those used in heplot(), but is otherwise the same.

Rotating the plot can be particularly revealing, showing views in which $\mathbf{H}$ variation is particularly large or small in relation to $\mathbf{E}$ variation. If you find a view making $\mathbf{H}$ as large as possible, this corresponds closely to the canonical view chosen by the candisc.

See rgl::play3d() and rgl::movie3d() for details on creating animations, rotating the 3D scene around on or more coordinate axes, or zooming in/out to highlight some feature.

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

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

Use in documents:

For use in an R Markdown, Quarto, or pkgdown document, follow the call to heplot3d() with rgl::rglwidget() to embed the plot as an interactive, rotatable widget instead of a static image.

- In a knitted document, call rgl::setupKnitr() with autoprint = TRUE once, near the top of the document, so that low-level rgl calls are captured automatically;
- On a pkgdown reference page like this, the trailing rgl::rglwidget() call in the example is enough by itself, and no setupKnitr() call is needed.

Note that the widget capture depends on rglwidget() itself being the visibly-printed value – heplot3d() returns its ellipsoid bounding boxes invisibly (for compatibility with heplot()), so calling it alone does not trigger a plot in either context.

For a bookdown-based vignette with pkgdown: as_is: true in its YAML header (needed for \@ref() figure/table numbering), calling rglwidget() directly renders fine in the standalone vignette (R CMD build/devtools::build_vignettes()) but shows as a blank box on the pkgdown site: pkgdown replaces the document's own <head> when assembling the article page, silently dropping the <script> tags the widget needs to render, even though its scene data is still present in the page. The fix is to pre-render the widget to a self-contained file and embed that with an <iframe> instead, since body content (unlike the <head>) survives pkgdown's page assembly:

```
w <- rgl::rglwidget()
htmlwidgets::saveWidget(w, "vignettes/images/myplot.html", selfcontained = TRUE)
```

and in the vignette text, in place of the live call:

```
htmltools::tags$iframe(src = "images/myplot.html", width = "100%", height = "500",
                       style = "border: none;")
```

pkgdown copies the referenced file into the built site automatically, the same way it already copies images referenced via knitr::include_graphics().

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. https://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

car::Anova(), car::linearHypothesis(), for details on MANOVA tests and linear hypotheses

heplot(), pairs.mlm(), for other plotting methods for mlm objects

rgl::rgl-package, for details about 3D plots with rgl

candisc::heplot3d.candisc() for 3D HE plots in canonical space.

Other HE plot functions: heplot(), heplot1d(), pairs.mlm()

Other 3D plotting: arrow3d(), bbox3d(), cross3d(), ellipse3d.axes()

Examples

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

# Plastic data
```

```
plastic.mod <- lm(cbind(tear, gloss, opacity) ~ rate*additive, data=Plastic)
## Not run:
heplot3d(plastic.mod, col=c("red", "blue", "brown", "green3"), wire=FALSE)

## End(Not run)
```

`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.

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` preoperative complications with heart, 1-4 scale, with 1=none, 2=mild, 3=moderate, 4=severe.

`resp` preoperative complications with respiration, 1-4 scale, 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. Data Exhibit 8, 567-568. Their source: A study by B. McPeek 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

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

# test overall regression
print(linearHypothesis(Hern.mod, c("age", "sexm", "pstat", "build", "cardiac", "resp")), SSP=FALSE)

# joint test of age, sex & caridac
print(linearHypothesis(Hern.mod, c("age", "sexm", "cardiac")), SSP=FALSE)

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

## Enhancing the pairs plot ...
# create better variable labels
vlab <- c("LeaveCondition\n(leave)",
          "NursingCare\n(nurse)",
          "LengthOfStay\n(los)")
# Add ellipse to test all 5 regressors simultaneously
hyp <- list("Regr" = c("age", "sexm", "pstat", "build", "cardiac", "resp"))
pairs(Hern.mod, hypotheses=hyp, col=clr, var.labels=vlab)

## Views in canonical space for the various predictors
if (require(candisc)) {
 Hern.canL <- candiscList(Hern.mod)
 plot(Hern.canL, term="age")
 plot(Hern.canL, term="sex")
 plot(Hern.canL, term="pstat")  # physical status
}
```

interpPlot *Plot an Interpolation Between Two Related Data Sets*

Description

Plot an interpolation between two related data sets, typically transformations of each other. This function is designed to be used in animations.

Usage

```
interpPlot(
  xy1,
  xy2,
  alpha,
  xlim,
  ylim,
  points = TRUE,
  add = FALSE,
  col = palette()[1],
  ellipse = FALSE,
  ellipse.args = NULL,
  abline = FALSE,
  col.lines = palette()[2],
  lwd = 2,
  id.method = "mahal",
  labels = rownames(xy1),
  id.n = 0,
  id.cex = 1,
  id.col = palette()[1],
  segments = FALSE,
  segment.col = "darkgray",
  ...
)
```

Arguments

xy1	First data set, a 2-column matrix or data.frame
xy2	Second data set, a 2-column matrix or data.frame
alpha	The value of the interpolation fraction, typically (but not necessarily) `0 <= alpha <= 1`.
xlim, ylim	x, y limits for the plot. If not specified, the function uses the ranges of `rbind(xy1, xy2)`.
points	Logical. Whether to plot the points in the current interpolation?
add	Logical. Whether to add to an existing plot?
col	Color for plotted points.
ellipse	logical. `TRUE` to plot a `dataEllipse`

ellipse.args	other arguments passed to dataEllipse
abline	logical. TRUE to plot the linear regression line for XY
col.lines	line color
lwd	line width
id.method	How points are to be identified. See car::showLabels().
labels	observation labels
id.n	Number of points to be identified. If set to zero, no points are identified.
id.cex	Controls the size of the plotted labels. The default is 1
id.col	Controls the color of the plotted labels.
segments	logical. TRUE to draw lines segments from xy1 to xy
segment.col	line color for segments
...	other arguments passed to plot()

Details

Points are plotted via the linear interpolation,

$$XY = XY1 + \alpha(XY2 - XY1)$$

The function allows plotting of the data ellipse, the linear regression line, and line segments showing the movement of points.

Interpolations other than linear can be obtained by using a non-linear series of alpha values. For example alpha=sin(seq(0,1,.1)/sin(1) will give a sinusoid interpolation.

Value

Returns invisibly the interpolated XY points.

Note

The examples here just use on-screen animations to the console graphics window. The animation::animation package provides facilities to save these in various formats.

Author(s)

Michael Friendly

See Also

car::dataEllipse(), car::showLabels(), animation::animation

Examples

```
################################################
# animate an AV plot from marginal to conditional
################################################
data(Duncan, package="carData")
duncmod <- lm(prestige ~ income + education, data=Duncan)
mod.mat <- model.matrix(duncmod)

# function to do an animation for one variable
dunc.anim <- function(variable, other, alpha=seq(0, 1, .1)) {
  var <- which(variable==colnames(mod.mat))
  duncdev <- scale(Duncan[,c(variable, "prestige")], scale=FALSE)
  duncav <- lsfit(mod.mat[, -var], cbind(mod.mat[, var], Duncan$prestige),
          intercept = FALSE)$residuals
  colnames(duncav) <- c(variable, "prestige")

  lims <- apply(rbind(duncdev, duncav),2,range)

  for (alp in alpha) {
    main <- if(alp==0) paste("Marginal plot:", variable)
      else paste(round(100*alp), "% Added-variable plot:", variable)
    interpPlot(duncdev, duncav, alp, xlim=lims[,1], ylim=lims[,2], pch=16,
      main = main,
      xlab = paste(variable, "| ", alp, other),
      ylab = paste("prestige | ", alp, other),
      ellipse=TRUE, ellipse.args=(list(levels=0.68, fill=TRUE, fill.alpha=alp/2)),
      abline=TRUE, id.n=3, id.cex=1.2, cex.lab=1.25)
    Sys.sleep(1)
  }
}

# show these in the R console
if(interactive()) {
dunc.anim("income", "education")

dunc.anim("education", "income")
}

##########################################
# correlated bivariate data with 2 outliers
# show rotation from data space to PCA space
##########################################

set.seed(123345)
x <- c(rnorm(100), 2, -2)
y <- c(x[1:100] + rnorm(100), -2, 2)
XY <- cbind(x=x, y=y)
rownames(XY) <- seq_along(x)
XY <- scale(XY, center=TRUE, scale=FALSE)

# start, end plots
```

```
car::dataEllipse(XY, pch=16, levels=0.68, id.n=2)
mod <- lm(y~x, data=as.data.frame(XY))
abline(mod, col="red", lwd=2)

pca <- princomp(XY, cor=TRUE)
scores <- pca$scores
car::dataEllipse(scores, pch=16, levels=0.68, id.n=2)
abline(lm(Comp.2 ~ Comp.1, data=as.data.frame(scores)), lwd=2, col="red")

# show interpolation

# functions for labels, as a function of alpha
main <- function(alpha) {if(alpha==0) "Original data"
  else if(alpha==1) "PCA scores"
  else paste(round(100*alpha,1), "% interpolation")}
xlab <- function(alpha) {if(alpha==0) "X"
  else if(alpha==1) "PCA.1"
  else paste("X +", alpha, "(X - PCA.1)")}
ylab <- function(alpha) {if(alpha==0) "Y"
  else if(alpha==1) "PCA.2"
  else paste("Y +", alpha, "(Y - PCA.2)")}

interpPCA <- function(XY, alpha = seq(0,1,.1)) {
  XY <- scale(XY, center=TRUE, scale=FALSE)
  if (is.null(rownames(XY))) rownames(XY) <- 1:nrow(XY)
  pca <- princomp(XY, cor=TRUE)
  scores <- pca$scores

  for (alp in alpha) {
    interpPlot(XY, scores, alp,
      pch=16,
      main = main(alp),
      xlab = xlab(alp),
      ylab = ylab(alp),
      ellipse=TRUE, ellipse.args=(list(levels=0.68, fill=TRUE, fill.alpha=(1-alp)/2)),
      abline=TRUE, id.n=2, id.cex=1.2, cex.lab=1.25, segments=TRUE)
    Sys.sleep(1)
  }
}

# show in R console
if(interactive()) {
interpPCA(XY)
}

## Not run:
library(animation)
saveGIF({
  interpPCA(XY, alpha <- seq(0,1,.1))},
  movie.name="outlier-demo.gif", ani.width=480, ani.height=480, interval=1.5)


## End(Not run)
```

`Iwasaki_Big_Five` *Personality Traits of Cultural Groups*

Description

This dataset, from Grice & Iwasaki (2007), gives scores on the five personality scales of the NEO PI-r (Costa & McCrae, 1992), called the "Big Five" personality traits: Neuroticism, Extraversion, Openness-to-Experience, Agreeableness, and Conscientiousness.

Format

A data frame with 203 observations on the following 7 variables.

`ID` ID number

`Group` a factor with levels `Eur` `Asian_Amer` `Asian_Intl`

`N` Neuroticism score

`E` Extraversion score

`O` Openness score

`A` Agreeableness score

`C` Conscientiousness score

Details

The groups are:

Eur European Americans (Caucasians living in the United States their entire lives)

Asian_Amer Asian Americans (Asians living in the United States since before the age of 6 years)

Asian_Intl Asian Internationals (Asians who moved to the United States after their 6th birthday)

The factor `Group` is set up to compare E vs. Asian and the two Asian groups

Source

Grice, J., & Iwasaki, M. (2007). A truly multivariate approach to MANOVA. *Applied Multivariate Research*, **12**, 199-226. doi:10.22329/amr.v12i3.660.

References

Costa Jr, P. T., & McCrae, R. R. (1992). *Revised NEO Personality Inventory (NEO PI-R) and NEO Five-Factor Inventory (NEOFFI) professional manual*. Psychological Assessment Resources.

Examples

```
data(Iwasaki_Big_Five)
# use Helmert contrasts for groups
contrasts(Iwasaki_Big_Five$Group) <-
   matrix(c(2, -1, -1,
            0, -1,  1), ncol=2)

str(Iwasaki_Big_Five)

Big5.mod <- lm(cbind(N, E, O, A, C) ~ Group, data=Iwasaki_Big_Five)
coef(Big5.mod)

car::Anova(Big5.mod)

# test contrasts
car::linearHypothesis(Big5.mod, "Group1", title = "Eur vs Asian")
car::linearHypothesis(Big5.mod, "Group2", title = "Asian: Amer vs Inter")

# heplots
labs <- c("Neuroticism", "Extraversion", "Openness", "Agreeableness", "Conscientiousness" )

heplot(Big5.mod,
       fill = TRUE, fill.alpha = 0.2,
       cex.lab = 1.5,
       xlab = labs[1], ylab = labs[2])

heplot(Big5.mod, variables = c(2,5),
       fill = TRUE, fill.alpha = 0.2,
       cex.lab = 1.5,
       xlab = labs[2], ylab = labs[5])

pairs(Big5.mod,
      fill = TRUE, fill.alpha = 0.2, var.labels = labs)


# canonical discriminant analysis
if (require(candisc)) {
library(candisc)
Big5.can <- candisc(Big5.mod)
Big5.can
heplot(Big5.can, fill = TRUE, fill.alpha = 0.1)
}
```

`label.ellipse` *Label an Ellipse in a Plot*

Description

`label.ellipse` is used to a draw text label on an ellipse at its center or somewhere around the periphery in a very flexible way. It is used in `heplot()`, `covEllipses()`, and `coefplot.mlm()`, but is also useful as a utility when plotting ellipses in base R graphics.

Usage

```
label.ellipse(
  ellipse,
  label,
  col = "black",
  label.pos = NULL,
  xpd = TRUE,
  tweak = 0.5 * c(strwidth("M"), strheight("M")),
  ...
)
```

Arguments

ellipse	A two-column matrix of coordinates for the ellipse boundary, for example as computed by car::ellipse().
label	Character string to be used as the ellipse label
col	Label color
label.pos	Label position relative to the ellipse. See details
xpd	Should the label be allowed to extend beyond the plot limits?
tweak	A vector of two lengths used to tweak label positions. Only used for label positions 1:4 or corresponding character or compass directions. The defaults are 0.5 times the height and width of the character "M" added or subtracted to the calculated (x, y) values.
...	Other parameters passed to graphics::text(), e.g., cex, col, ...

Details

The function takes the coordinates of the input ellipse and uses that, together with label.pos to calculate the (x, y) coordinates to be passed to graphics::text() along with a computed pos argument. The values of tweak are applied to (x, y) to position the labels to the outside of the ellipse by default.

The label.pos argument implements a very general way to position the text label with respect to the ellipse:

- If label.pos = NULL (the default), the function uses the sign of the correlation r represented by the ellipse to determine a position at the "top" ($r >= 0$) or "bottom" ($r < 0$) of the ellipse.
- Integer values of 0, 1, 2, 3 and 4, respectively indicate positions at the center, below, to the left of, above and to the right of the max/min coordinates of the ellipse, where the values 1:4 correspond to the usual values of pos in graphics::text().
- Label positions can also be specified as the corresponding character strings c("center", "bottom", "left", "top", "right"), or *compass directions*, c("C", "S", "W", "N", "E"). Additionally, *diagonal* compass directions c("NE", "SE", "SW", "NW") can be used, corresponding to angles 45, 135, 225, and 315 degrees, clockwise from 0 at North.
- Even more generally, label.pos can also be a *fraction* in (0,1), interpreted as the fraction of the way around the unit circle, counterclockwise from the East point (1, 0). Thus, 0.25 is the top, 0.5 is the left, and 0.75 is the bottom.

Value

Mainly used for its side-effect of producing a call to graphics::text(), but also returns, invisibly, the (x, y) coordinates where the label was placed.

Author(s)

Michael Friendly

See Also

graphics::text(), car::ellipse(), heplot(), covEllipses()

Examples

```
# Helper, to compute a circle
circle <- function(center=c(0,0), radius=1, segments=60) {
  angles <- (0:segments)*2*pi/segments
  circle <- radius * cbind( cos(angles), sin(angles))
  t( c(center) + t( circle ))
}

# Create a circular ellipse
circ <- circle(radius=1.5)

plot(-2:2, -2:2,
     type="n", asp=1,
     main="Compass Directions on Circle\n(Cardinal + Diagonal)")
lines(circ, col="gray", lwd=2)
points(0, 0, pch="+", cex=2)

# Cardinal directions
cardinal <- c("N", "E", "S", "W")
for (pos in cardinal) {
  label.ellipse(circ, label=pos, label.pos=pos, col="red", cex=1.2, font=2)
}

# Diagonal directions (recently added)
diagonal <- c("NE", "SE", "SW", "NW")
for (pos in diagonal) {
  label.ellipse(circ, label=pos, label.pos=pos, col="blue", cex=1.2, font=2)
}

# Center, & illustrate return
xy <- label.ellipse(circ, label="C", label.pos="C", col="darkgreen", cex=1.2, font=2)
xy

# Add reference lines to show the angles
abline(h=0, v=0, col="lightgray", lty=2)
abline(a=0, b=1, col="lightgray", lty=2)   # 45° line
abline(a=0, b=-1, col="lightgray", lty=2)  # -45° line

legend("bottomleft",
```

```
        legend=c("Cardinal (N,E,S,W)", "Diagonal (NE,SE,SW,NW)", "Center"),
        col=c("red", "blue", "darkgreen"),
        lwd=2,  bty="n")

# Use in `heplot()`
data(dogfood, package = "heplots")
dogfood.mod <- lm(cbind(start, amount) ~ formula, data=dogfood)

# default: top or bottom, depending on sign of the ellipse
heplot(dogfood.mod,
       fill = TRUE, fill.alpha = 0.1)

# change label.pos and cex
heplot(dogfood.mod,
       fill = TRUE, fill.alpha = 0.1,
       label.pos = c("NE", "SW"), cex = 1.4)


# Define diagonal compass positions and their corresponding angular fractions
# translate nmemonics to standard numerical text positions 1:4,
```

`LearnDis` *Reading and Arithmetic Achievement in Children with Learning Disabilities*

Description

A small factorial dataset from Tabachnick & Fidell (2013) relating a treatment/control intervention and degree of disability to reading and arithmetic achievement test scores, with IQ also recorded. It is not clear whether this is fictitious data or derived from a real study.

It provides for simple examples of MANOVA, MANCOVA and stepdown analysis

Usage

```
data("LearnDis")
```

Format

A data frame with 18 observations on the following 5 variables, a 3 (`Disability`) x 2 (`Treatment`) between-subjects factorial with n = 3 per cell.

`Disability` Degree of disability, an ordered factor with levels `Mild` < `Moderate` < `Severe`

`Treatment` a factor with levels `Treatment` `Control`

`WRAT_R` Wide Range Achievement Test, Reading subtest score, a numeric vector

`WRAT_A` Wide Range Achievement Test, Arithmetic subtest score, a numeric vector

`IQ` IQ score, a numeric vector – used in the source as a MANCOVA covariate, not part of the stepdown analysis itself

Source

Tabachnick, B. G., & Fidell, L. S. (2013). *Using Multivariate Statistics* (6th ed.). Pearson. Table 7.1, p.256.

References

The Roy-Bargmann stepdown analysis of this data (WRAT-R prioritized over WRAT-A) appears in the same source, §7.5.3.2, Tables 7.7-7.9, pp.273-274.

Roy, S. N. (1958). Step-Down Procedure in Multivariate Analysis. *The Annals of Mathematical Statistics*, 29(4), 1177-1187. doi:10.1214/aoms/1177706449.

Examples

```
data(LearnDis)
str(LearnDis)

ld.mod <- lm(cbind(WRAT_R, WRAT_A) ~ Disability * Treatment, data = LearnDis)
car::Anova(ld.mod)

heplot(ld.mod, fill = TRUE, fill.alpha = 0.1)

# Roy-Bargmann stepdown: does WRAT-A add anything to WRAT-R for the
# Treatment effect?
# Needs Type III SS, since the WRAT_R covariate breaks the balanced
# factorial's orthogonality:
options(contrasts = c("contr.sum", "contr.poly"))
step2.mod <- lm(WRAT_A ~ WRAT_R + Disability * Treatment, data = LearnDis)
car::Anova(step2.mod, type = "III")
```

---

leveneTests *Levene Tests of Homogeneity of Variances*

Description

This function extends `car::leveneTest()` to a multivariate response setting. It performs the Levene test of homogeneity of variances for each of a set of response variables, and prints a compact summary.

Usage

```
leveneTests(y, ...)

## Default S3 method:
leveneTests(y, group, center = median, ...)

## S3 method for class 'formula'
leveneTests(y, data, ...)
```

```
## S3 method for class 'lm'
leveneTests(y, ...)
```

Arguments

y	A data frame or matrix of numeric response variables for the default method, or a model formula for a multivariate linear model, or the multivariate linear model itself. In the case of a formula or model, the variables on the right-hand-side of the model must all be factors and must be completely crossed.
...	arguments to be passed down to car::leveneTest(), e.g., data for the formula and lm methods; can also be used to pass arguments to the function given by center (e.g., center=mean and trim=0.1 specify the 10% trimmed mean) other arguments.
group	a vector or factor object giving the group for the corresponding elements of the rows of y for the default method
center	The name of a function to compute the center of each group; mean gives the original Levene's (1960) test; the default, median, provides a more robust test suggested by Brown and Forsythe (1974).
data	the data set, for the formula method

Value

An object of classes "anova" and "data.frame", with one observation for each response variable in y.

Author(s)

Michael Friendly

References

Levene, H. (1960). Robust Tests for Equality of Variances. In Olkin, I. *et al.* (Eds.), *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, Stanford University Press, 278-292.

Brown, M. B. & Forsythe, A. B. (1974). Robust Tests For Equality Of Variances *Journal of the American Statistical Association*, **69**, 364-367.

See Also

car::leveneTest(), bartlettTests()

Other homogeneity tests: bartlettTests()

Examples

```
leveneTests(iris[,1:4], iris$Species)

# handle a 1-column response?
```

```
leveneTests(iris[,1, drop=FALSE], iris$Species)

data(Skulls, package="heplots")
leveneTests(Skulls[,-1], Skulls$epoch)

# formula method
leveneTests(cbind(mb, bh, bl, nh) ~ epoch, data=Skulls)

# use 10% trimmed means
leveneTests(cbind(mb, bh, bl, nh) ~ epoch, data=Skulls, trim = 0.1)


# mlm method
skulls.mod <- lm(cbind(mb, bh, bl, nh) ~ epoch, data=Skulls)
leveneTests(skulls.mod)
```

`logdetCI` *Calculate confidence interval for log determinant of covariance matrices*

Description

This function uses asymptotic results described by Cai et. al (2016), Theorem 1, to calculate approximate, normal theory confidence intervals (CIs) for the log determinant of one or more sample covariance matrices.

Usage

```
logdetCI(cov, n, conf = 0.95, method = 1, bias.adj = TRUE)
```

Arguments

`cov`	a covariance matrix or a (named) list of covariance matrices, all the same size
`n`	sample size, or vector of sample sizes, one for each covariance matrix
`conf`	confidence level
`method`	Three methods are provided, based on Cai et. al Theorem 1 (`method=1`), Corollary 1 (`method=2`) and Corollary 2 (`method=3`), each with different bias and SE values.
`bias.adj`	logical; set `FALSE` to exclude the bias correction term.

Details

Their results are translated into a CI via the approximation

$$\log det(\widehat{\Sigma}) - bias \pm z_{1-\alpha/2} \times SE$$

where $\widehat{\Sigma}$ is the sample estimate of a population covariance matrix, $bias$ is a bias correction constant and SE is a width factor for the confidence interval. Both $bias$ and SE are functions of the sample size, n and number of variables, p.

This function is included here only to provide an approximation to *graphical accuracy* for use with Box's M test for equality of covariance matrices, `boxM()` and its associated `plot.boxM()` method.

Cai et. al (2015) claim that their Theorem 1 holds with either p fixed or $p(n)$ growing with n, as long as $p(n) \leq n$. Their Corollary 1 (`method=2`) is the special case when p is fixed. Their Corollary 2 (`method=3`) is the special case when $0 \leq p/n < 1$ is fixed.

The properties of this CI estimator are unknown in small to moderate sample sizes, but it seems to be the only one available. It is therefore experimental in this version of the package and is subject to change in the future.

The $bias$ term offsets the confidence interval from the sample estimate of $\log det(\widehat{\Sigma})$. When p is large relative to n, the confidence interval may not overlap the sample estimate.

Strictly speaking, this estimator applies to the MLE of the covariance matrix $\widehat{\Sigma}$, i.e., using n rather than $n-1$ in as the divisor. The factor $(n-1/n)$ has not yet been taken into account here.

Value

A data frame with one row for each covariance matrix. `lower` and `upper` are the boundaries of the confidence intervals. Other columns are `logdet, bias, se`.

Author(s)

Michael Friendly

References

Cai, T. T.; Liang, T. & Zhou, H. H. (2015) Law of log determinant of sample covariance matrix and optimal estimation of differential entropy for high-dimensional Gaussian distributions. *Journal of Multivariate Analysis*, 137, 161-172. doi:10.1016/j.jmva.2015.02.003

See Also

`boxM()`, `plot.boxM()`

Examples

```
data(iris)
iris.mod <- lm(as.matrix(iris[,1:4]) ~ iris$Species)
iris.boxm <- boxM(iris.mod)
cov <- c(iris.boxm$cov, list(pooled=iris.boxm$pooled))
n <- c(rep(50, 3), 150)

CI <- logdetCI( cov, n=n, conf=.95, method=1)
CI
plot(iris.boxm, xlim=c(-14, -8), main="Iris data, Box's M test", gplabel="Species")
arrows(CI$lower, 1:4, CI$upper, 1:4, lwd=3, angle=90, len=.1, code=3)

CI <- logdetCI( cov, n=n, conf=.95, method=1, bias.adj=FALSE)
```

```
CI
plot(iris.boxm, xlim=c(-14, -8), main="Iris data, Box's M test", gplabel="Species")
arrows(CI$lower, 1:4, CI$upper, 1:4, lwd=3, angle=90, len=.1, code=3)
```

`Mahalanobis` *Classical and Robust Mahalanobis Distances*

Description

This function is a convenience wrapper to `stats::mahalanobis()` offering also the possibility to calculate robust Mahalanobis squared distances using MCD and MVE estimators of center and covariance (from `MASS::cov.rob()`)

Usage

```
Mahalanobis(
  x,
  center,
  cov,
  method = c("classical", "mcd", "mve"),
  nsamp = "best",
  ...
)
```

Arguments

`x`	a numeric matrix or data frame with, say, p columns
`center`	mean vector of the data; if this and `cov` are both supplied, the function simply calls `stats::mahalanobis()` to calculate the result
`cov`	covariance matrix (p x p) of the data
`method`	estimation method used for center and covariance, one of: `"classical"` (product-moment), `"mcd"` (minimum covariance determinant), or `"mve"` (minimum volume ellipsoid).
`nsamp`	passed to `MASS::cov.rob()`
`...`	other arguments passed to `MASS::cov.rob()`

Details

Any missing data in a row of `x` causes `NA` to be returned for that row.

Value

```
a vector of length `nrow(x)` containing the squared distances.
```

Author(s)

Michael Friendly

See Also

`stats::mahalanobis()`, `MASS::cov.rob()`

Other robust methods: `plot.robmlm()`, `robmlm()`

Examples

```
summary(Mahalanobis(iris[, 1:4]))
summary(Mahalanobis(iris[, 1:4], method="mve"))
summary(Mahalanobis(iris[, 1:4], method="mcd"))
```

---

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 size=5*cex in a call to rgl::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 lty=0.
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()
```

---

mathscore *Math scores for basic math and word problems*

Description

Scores for two groups of school children taught by different math teachers and tested for both basic math (BM) problems and solving word problems (WP).

Format

A data frame with 12 observations on the following 3 variables.

group a factor with levels 1 2

BM Basic Math score, a numeric vector

WP Word Problems score, a numeric vector

Source

Fictitious data

Examples

```
data(mathscore)
str(mathscore)

math.mod <- lm(cbind(BM, WP) ~ group, data=mathscore)
car::Anova(math.mod)

# scatterplot with data ellipses
car::scatterplot(WP ~ BM | group, data=mathscore,
 ellipse=list(levels=0.68), smooth=FALSE, pch=c(15,16),
 legend=list(coords = "topright"))

# HE plot
heplot(math.mod, fill=TRUE,
  cex=2, cex.lab=1.8,
 xlab="Basic math", ylab="Word problems")
```

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.

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'

Details

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

Source

Originally obtained from Dr. Wuensch's StatData page at East Carolina University. No longer exists. % was: https://core.ecu.edu/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)
car::Anova(jury.mod1, test="Roy")

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

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

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

NeuroCog *Neurocognitive Measures in Psychiatric Groups*

Description

The primary purpose of the study (Hartman, 2016, Heinrichs et al. (2015)) was to evaluate patterns and levels of performance on neurocognitive measures among individuals with schizophrenia and schizoaffective disorder using a well-validated, comprehensive neurocognitive battery specifically designed for individuals with psychosis (Heinrichs et al. (2008))

Format

A data frame with 242 observations on the following 10 variables.

`Dx` Diagnostic group, a factor with levels `Schizophrenia` `Schizoaffective` `Control`

`Speed` Speed of processing domain T score, a numeric vector

`Attention` Attention/Vigilance Domain T score, a numeric vector

`Memory` Working memory a numeric vector

`Verbal` Verbal Learning Domain T score, a numeric vector

`Visual` Visual Learning Domain T score, a numeric vector

`ProbSolv` Reasoning/Problem Solving Domain T score, a numeric vector

`SocialCog` Social Cognition Domain T score, a numeric vector

`Age` Subject age, a numeric vector

`Sex` Subject gender, a factor with levels `Female` `Male`

Details

The main interest was in determining how well these measures distinguished among all groups and whether there were variables that distinguished between the schizophrenia and schizoaffective groups.

Neurocognitive function was assessed using the MATRICS Consensus Cognitive Battery (MCCB; Nuechterlein et al., 2008). The MCCB consists of 10 individually administered tests that measure cognitive performance in seven domains: speed of processing, attention/vigilance, working memory, verbal learning, visual learning, reasoning and problem solving, and social cognition.

The clinical sample comprised 116 male and female patients who met the following criteria: 1) a diagnosis of schizophrenia (n = 70) or schizoaffective disorder (n = 46) confirmed by the Structured Clinical Interview for DSM-IV-TR Axis I Disorders; 2) outpatient status; 3) a history free of

developmental or learning disability; 4) age 18-65; 5) a history free of neurological or endocrine disorder; and 6) no concurrent DSM-IV-TR diagnosis of substance use disorder.

Non-psychiatric control participants (n = 146) were screened for medical and psychiatric illness and history of substance abuse. Patients were recruited from three outpatient clinics in Hamilton, Ontario, Canada. Control participants were recruited through local newspaper and online classified advertisements for paid research participation.

Source

Hartman, L. I. (2016). Schizophrenia and Schizoaffective Disorder: One Condition or Two? Unpublished PhD dissertation, York University.

Heinrichs, R.W., Pinnock, F., Muharib, E., Hartman, L.I., Goldberg, J.O., & McDermid Vaz, S. (2015). Neurocognitive normality in schizophrenia revisited. *Schizophrenia Research: Cognition*, **2** (4), 227-232. doi: 10.1016/j.scog.2015.09.001

References

Heinrichs, R. W., Ammari, N., McDermid Vaz, S. & Miles, A. (2008). Are schizophrenia and schizoaffective disorder neuropsychologically distinguishable? *Schizophrenia Research*, **99**, 149-154.

Nuechterlein K.H., Green M.F., Kern R.S., Baade L.E., Barch D., Cohen J., Essock S., Fenton W.S., Frese F.J., Gold J.M., Goldberg T., Heaton R., Keefe R.S.E., Kraemer H., Mesholam-Gately R., Seidman L.J., Stover E., Weinberger D.R., Young A.S., Zalcman S., Marder S.R. (2008) The MATRICS Consensus Cognitive Battery, Part 1: Test selection, reliability, and validity. *American Journal of Psychiatry*, **165** (2), 203-213. doi:10.1176/appi.ajp.2007.07010042.

Examples

```
library(car)
data(NeuroCog)
NC.mlm <- lm(cbind( Speed, Attention, Memory, Verbal, Visual, ProbSolv) ~ Dx,
               data=NeuroCog)
Anova(NC.mlm)

# test contrasts
contrasts(NeuroCog$Dx)
print(linearHypothesis(NC.mlm, "Dx1"), SSP=FALSE)
print(linearHypothesis(NC.mlm, "Dx2"), SSP=FALSE)

# pairwise HE plots
pairs(NC.mlm, var.cex=1.5)

# canonical discriminant analysis
if (require(candisc)) {
  NC.can <- candisc(NC.mlm)
  NC.can

  plot(NC.can, ellipse=TRUE, rev.axes=c(TRUE,FALSE), pch=c(7,9,10))
}
```

NLSY *National Longitudinal Survey of Youth Data*

Description

The dataset NLSY comes from a small part of the National Longitudinal Survey of Youth, which is a series of annual surveys conducted by the U.S. Department of Labor to examine the transition of young people into the labor force. This particular subset gives measures of 243 children on mathematics and reading achievement and also measures of behavioral problems (antisocial, hyperactivity). Also available are the yearly income and education of the child's father.

Format

A data frame with 243 observations on the following 6 variables.

`math` Math achievement test score

`read` Reading achievement test score

`antisoc` score on a measure of child's antisocial behavior, `0:6`

`hyperact` score on a measure of child's hyperactive behavior, `0:5`

`income` yearly income of child's father

`educ` years of education of child's father

Details

For the examples using this dataset, `math` and `read` scores are taken at the outcome variables. Among the remaining predictors, `income` and `educ` might be considered as background variables necessary to control for. Interest might then be focused on whether the behavioral variables `antisoc` and `hyperact` contribute beyond that.

The distribution of father's income is very highly skewed in the positive direction. Linear model analysis should probably use `log(income)`, but this is omitted for simplicity.

The dataset also contains a few unusual observations for you to discover.

Source

This dataset was derived from a larger one used by Patrick Curran at the 1997 meeting of the Society for Research on Child Development (SRCD). A description now only exists on the WayBack Machine, http://web.archive.org/web/20050404145001/http://www.unc.edu/~curran/example.html.

More details are available at: http://web.archive.org/web/20060830061414/http://www.unc.edu/~curran/srcd-docs/srcdmeth.pdf.

See also: The Index to the NLSY97 Cohort, https://www.nlsinfo.org/content/cohorts/nlsy97

Examples

```
library(car)
data(NLSY)

#examine the data
scatterplotMatrix(NLSY, smooth=FALSE)

# test control variables by themselves
# -------------------------------------
NLSY.mod1 <- lm(cbind(read, math) ~ income + educ, data=NLSY)
Anova(NLSY.mod1)
heplot(NLSY.mod1, fill=TRUE)

# test of overall regression
coefs <- rownames(coef(NLSY.mod1))[-1]
linearHypothesis(NLSY.mod1, coefs)
heplot(NLSY.mod1, fill=TRUE, hypotheses=list("Overall"=coefs))

# coefficient plot
coefplot(NLSY.mod1, fill = TRUE,
         col = c("darkgreen", "brown"),
         lwd = 2,
         ylim = c(-0.5, 3),
      main = "Bivariate coefficient plot for reading and math\nwith 95% confidence ellipses")


# additional contribution of antisoc + hyperact over income + educ
# -----------------------------------------------------------------
NLSY.mod2 <- lm(cbind(read,math) ~ antisoc + hyperact + income + educ, data=NLSY)
Anova(NLSY.mod2)

coefs <- rownames(coef(NLSY.mod2))[-1]
heplot(NLSY.mod2, fill=TRUE, hypotheses=list("Overall"=coefs, "mod2|mod1"=coefs[1:2]))
linearHypothesis(NLSY.mod2, coefs[1:2])

heplot(NLSY.mod2, fill=TRUE, hypotheses=list("mod2|mod1"=coefs[1:2]))

# check for outliers
idx <- cqplot(NLSY.mod2, id.n = 5)
idx
```

`noteworthy` *Find noteworthy (unusual) points in a 2D plot*

Description

This function extends the logic used by `car::showLabels()` to provide a more general collection of methods to identify unusual or "noteworthy" points in a two-dimensional display. Standard methods include Mahalanobis and Euclidean distance from the centroid, absolute value of distance from the mean of X or Y, absolute value of Y and absolute value of the residual in a model `Y ~ X`.

Usage

```
noteworthy(x, y = NULL, n = length(x), method = "mahal", level = NULL, ...)
```

Arguments

x, y	The x and y coordinates of a set of points. Alternatively, a single argument x can be provided, since `[grDevices::xy.coords()](x, y)` is used for construction of the coordinates.
n	Maximum number of points to identify. If set to 0, no points are identified.
method	Method of point identification. See Details.
level	Where appropriate, if supplied, the identified points are filtered so that only those for which the criterion is < `level`
...	Other arguments, silently ignored

Details

The `method` argument determines how the points to be identified are selected:

`"mahal"` Treat (x, y) as if it were a bivariate sample, and select cases according to their Mahalanobis distance from `(mean(x), mean(y))`.

`"dsq"` Similar to `"mahal"` but uses squared Euclidean distance.

`"x"` Select points according to their value of `abs(x - mean(x))`.

`"y"` Select points according to their value of `abs(y - mean(y))`.

`"r"` Select points according to their value of `abs(y)`, as may be appropriate in residual plots, or others with a meaningful origin at 0, such as a chi-square QQ plot.

`"ry"` Fit the linear model, `y ~ x` and select points according to their absolute residuals.

case IDs `method` can be an integer vector of case numbers in `1:length{x}`, in which case those cases will be labeled.

numeric vector `method` can be a vector of the same length as x consisting of values to determine the points to be labeled. For example, for a linear model `mod`, setting `method=cooks.distance(mod)` will label the `n` points corresponding to the largest values of Cook's distance. Warning: If missing data are present, points may be incorrectly selected.

In the case of `method == "mahal"` a value for `level` can be supplied. This is used as a filter to select cases whose criterion value exceeds `level`. In this case, the number of points identified will be less than or equal to `n`.

Examples

```
# example code
set.seed(47)
x <- c(runif(100), 1.5, 1.6, 0)
y <- c(2*x[1:100] + rnorm(100, sd = 1.2), -2, 6, 6 )
z <- y - x
mod <- lm(y ~ x)

# testing function to compare noteworthy with car::showLabels()
```

```
testnote <- function(x, y, n, method=NULL, ...)  {
  plot(x, y)
  abline(lm(y ~ x))
  if (!is.null(method))
    car::showLabels(x, y, n=n, method = method) |> print()
  ids <- noteworthy(x, y, n=n, method = method, ...)
  text(x[ids], y[ids], labels = ids, col = "red")
  ids
  }

  # Mahalanobis distance
testnote(x, y, n = 5, method = "mahal")
testnote(x, y, n = 5, method = "mahal", level = .99)
  # Euclidean distance
testnote(x, y, n = 5, method = "dsq")

testnote(x, y, n = 5, method = "y")
testnote(x, y, n = 5, method = "ry")

  # a vector of criterion values
testnote(x, y, n = 5, method = Mahalanobis(data.frame(x,y)))
testnote(x, y, n = 5, method = z)
  # vector of case IDs
testnote(x, y, n = 4, method = seq(10, 60, 10))
testnote(x, y, n = 4, method = which(cooks.distance(mod) > .25))

  # test use of xy.coords
noteworthy(data.frame(x,y), n=4)
noteworthy(y ~ x, n=4)
```

`oral` *Effect of Delay in Oral Practice in Second Language Learning*

Description

Postovsky (1970) investigated the effect of delay in oral practice at the beginning of second language learning. A control condition began oral practice with no delay, while an experimental group had a four-week delay before starting oral practice. The data consists of scores on language skills at the end of six weeks of study.

Students in this study were matched on age, education, former language training, intelligence and language aptitude.

Usage

```
data("oral")
```

Format

A data frame with 56 observations on the following 5 variables.

`group` Group, a factor with levels `Control` `Exptl`

`listen` Listening test, a numeric vector

`speak` Speaking test, a numeric vector

`read` Reading test, a numeric vector

`write` Writing test, a numeric vector

Source

Timm, N. H. (1975). *Multivariate Analysis with Applications in Education and Psychology*. Wadsworth (Brooks/Cole), Exercise 3.12, p. 279.

References

Postovsky, V. A. (1970). Effects of delay in oral practice at the start of second language training. Unpublished doctoral dissertation, University of California, Berkeley.

Examples

```
library(car)
library(candisc)
data(oral)

# make some boxplots
op <- par(mfrow=c(1,4), cex.lab=1.5)
clr <- c("pink", "lightblue")
Boxplot(listen ~ group, data=oral, col = clr, cex.lab = 1.5)
Boxplot(speak ~  group, data=oral, col = clr, cex.lab = 1.5)
Boxplot(read ~   group, data=oral, col = clr, cex.lab = 1.5)
Boxplot(write ~  group, data=oral, col = clr, cex.lab = 1.5)
par(op)

# view the data ellipses
covEllipses(cbind(listen, speak, read, write) ~ group, data=oral,
    variables = 1:4,
    level = 0.40,
    pooled = FALSE,
    fill = TRUE, fill.alpha = 0.05)

oral.mod <- lm(cbind(listen, speak, read, write) ~ group, data=oral)
Anova(oral.mod)

# canonical view
oral.can <- candisc(oral.mod) |> print()
summary(oral.can)

# reflect the structure & scores to make them positive
oral.can$structure[, "Can1"] <- -1 * oral.can$structure[, "Can1"]
```

```
oral.can$scores[, "Can1"]    <- -1 * oral.can$scores[, "Can1"]

plot(oral.can, var.lwd=2)
```

`Oslo` *Oslo Transect Subset Data*

Description

The Oslo data set contains chemical concentrations of 332 samples of different plant species collected along a 120 km transect running through the city of Oslo, Norway. It is a subset of the `rrcov::OsloTransect` data provided by the `rrcov` package.

Format

A data frame with 332 observations on the following 14 variables.

`site` transect site ID, a factor with levels `102` `103` `104` `105` `106` `107` `108` `109` `111` `112` `113` `114` `115` `116` `117` `118` `119` `121` `122` `123` `124` `125` `126` `127` `128` `129` `131` `132` `133` `134` `135` `136` `138` `139` `141` `142` `143` `144`

`XC` X coordinate, a numeric vector

`YC` Y coordinate, a numeric vector

`forest` forest type, a factor with levels `birspr` `mixdec` `pine` `sprbir` `sprpin` `spruce`

`weather` weather type, a factor with levels `cloud` `moist` `nice` `rain`

`litho` lithological type, a factor with levels `camsed` (Cambro-Silurian sedimentary), `gneis_o` (Precambrian gneisses - Oslo), `gneis_r` (- Randsfjord), `magm` (Magmatic rocks)

`altitude` altitude, a numeric vector

`Cu` Copper, a numeric vector

`Fe` Iron, a numeric vector

`K` Potassium, a numeric vector

`Mg` Magnesium, a numeric vector

`Mn` Manganese, a numeric vector

`P` Lead, a numeric vector

`Zn` Zinc, a numeric vector

Details

The `rrcov::OsloTransect` contains 360 observations, with 9 observations per site. Only 7 chemical elements were retained from the 25 contained in the `rrcov::OsloTransect` data, and these were all log-transformed, following Todorov and Filzmoser (2009).

Only complete cases on these variables were retained, and two lithological types of low frequency were removed, leaving 332 observations.

Source

Reimann, C., Arnoldussen, A., Boyd, R., Finne, T.E., Koller, F., Nordgulen, Oe., And Englmaier, P. (2007) Element contents in leaves of four plant species (birch, mountain ash, fern and spruce) along anthropogenic and geogenic concentration gradients, *The Science of the Total Environment*, 377, 416-433.

References

Todorov V. and Filzmoser P. (2009) Robust statistic for the one-way MANOVA, submitted to the *Journal of Environmetrics*.

Examples

```
data(Oslo)
table(Oslo$litho)

Oslo.mod <- lm(cbind(Cu, K, Mg, Mn, P, Zn) ~ litho, data=Oslo)
car::Anova(Oslo.mod)

heplot(Oslo.mod, var=c("Cu", "Mn"))
pairs(Oslo.mod)

## Not run:
if(require(candisc)) {
  Oslo.can <- candisc(Oslo.mod)
  Oslo.can
  heplot(Oslo.can)
  if(requireNamespace("rgl")){
    heplot3d(Oslo.can, shade=TRUE, wire=FALSE, alpha=0.5, var.col="red")
  }
}

## End(Not run)
```

Overdose *Overdose of Amitriptyline*

Description

Data on overdoses of the drug amitriptyline. Amitriptyline is a drug prescribed by physicians as an antidepressant. However, there are also conjectured side effects that seem to be related to the use of the drug: irregular heart beat, abnormal blood pressure and irregular waves on the electrocardiogram (ECG). This dataset (originally from Rudorfer, 1982) gives data on 17 patients admitted to hospital after an overdose of amitriptyline. The two response variables are: `TCAD` and `AMI`. The other variables are predictors.

Usage

```
data("Overdose")
```

Format

A data frame with 17 observations on the following 7 variables.

`TCAD` total TCAD plasma level, a numeric vector

`AMI` amount of amitriptyline present in the TCAD plasma level, a numeric vector

`Gender` a factor with levels `Male` `Female`

`amount` amount of drug taken at time of overdose, a numeric vector

`BP` diastolic blood pressure, a numeric vector

`ECG_PR` ECG PR wave measurement, a numeric vector

`ECG_QRS` ECG QRS wave measurement, a numeric vector

%% @details %% ~~ If necessary, more details than the **description** above ~~

Source

Johnson & Wichern (2005), *Applied Multivariate Statistical Analysis*, Exercise 7.25, p. 426.

References

Rudorfer, M. V. Cardiovascular changes and plasma drug levels after amitriptyline overdose. (1982). *J. Toxicology - Clinical Toxicology*. **19**(1),67-78. doi:10.3109/15563658208990367, PMID: 7154142.

Clay Ford, "Getting started with Multivariate Multiple Regression", https://library.virginia.edu/data/articles/getting-started-with-multivariate-multiple-regression.

ECG measurements:

PR https://en.wikipedia.org/wiki/PR_interval

QRS https://en.wikipedia.org/wiki/QRS_complex

Examples

```
data(Overdose)
str(Overdose)
pairs(Overdose)

over.mlm <- lm(cbind(TCAD, AMI) ~ Gender + amount + BP + ECG_PR + ECG_QRS, data = Overdose)
coef(over.mlm)

# check for outliers
cqplot(over.mlm)

# HE plot shows that relations of responses to predictors are essentially one-dimensional
heplot(over.mlm)

# canonical correlation analysis
```

```
if(require(candisc)) {
cancor(cbind(TCAD, AMI) ~ as.numeric(Gender) + amount + BP + ECG_PR + ECG_QRS, data = Overdose)
}
```

`pairs.mlm` *Pairwise HE Plots*

Description

The function (in the form of an `mlm` method for the generic `graphics::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,
  var.cex = 2,
  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

`x`	an object of class `mlm`.
`variables`	indices or names of the three of more response variables to be plotted; defaults to all of the responses.
`var.labels`	labels for the variables plotted in the diagonal panels; defaults to names of the response variables.
`var.cex`	character expansion for the variable labels.
`type`	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).

idata	an optional data frame giving a factor or factors defining the intra-subject model for multivariate repeated-measures data. See Details of car::Anova() for an explanation of the intra-subject design and for further explanation of the other arguments relating to intra-subject factors.
idesign	a one-sided model formula using the "data" in idata and specifying the intra-subject design for repeated measure models.
icontrasts	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 c("contr.sum", "contr.poly").
imatrix	In lieu of idata and idesign, you can specify the intra-subject design matrix directly via imatrix, 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 *different* terms must be mutually orthogonal.
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 iterm effect as well as all interactions of iterm with terms.
manova	optional Anova.mlm 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 heplot, which is used to draw each panel of the display.

Author(s)

Michael Friendly

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. https://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()

Other HE plot functions: heplot(), heplot1d(), 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")))
```

Parenting *Father Parenting Competence*

Description

The data, from an exercise given by Meyers et al. (2006) relates to 60 fathers assessed on three subscales of a Perceived Parenting Competence Scale. The fathers were selected from three groups: (a) fathers of a child with no disabilities; (b) fathers with a physically disabled child; (c) fathers with a mentally disabled child.

Format

A data frame with 60 observations on the following 4 variables.

`group` a factor with levels `Normal` `Physical Disability` `Mental Disability`

`caring` caretaking responsibilities, a numeric vector

`emotion` emotional support provided to the child, a numeric vector

`play` recreational time spent with the child, a numeric vector

Details

The scores on the response variables are discrete.

Source

Meyers, L. S., Gamst, G, & Guarino, A. J. (2006). *Applied Multivariate Research: Design and Interpretation*, Thousand Oaks, CA: Sage Publications, https://study.sagepub.com/meyers3e, Exercises 10B.

Examples

```
data(Parenting)
require(car)

# fit the MLM
parenting.mod <- lm(cbind(caring, emotion, play) ~ group, data=Parenting)
car::Anova(parenting.mod)

# Box's M test
boxM(parenting.mod)
plot(boxM(parenting.mod))

parenting.mod <- lm(cbind(caring, emotion, play) ~ group, data=Parenting)
car::Anova(parenting.mod)
# test contrasts
print(linearHypothesis(parenting.mod, "group1"), SSP=FALSE)
print(linearHypothesis(parenting.mod, "group2"), SSP=FALSE)


heplot(parenting.mod)
# display tests of contrasts
hyp <- list("N:MP" = "group1", "M:P" = "group2")
heplot(parenting.mod, hypotheses=hyp)

# make a prettier plot
heplot(parenting.mod, hypotheses=hyp, asp=1,
       fill=TRUE, fill.alpha=c(0.3,0.1),
       col=c("red", "blue"),
       lty=c(0,0,1,1), label.pos=c(1,1,3,2),
       cex=1.4, cex.lab=1.4, lwd=3)

pairs(parenting.mod, fill=TRUE, fill.alpha=c(0.3, 0.1))

## Not run:
heplot3d(parenting.mod, wire=FALSE)

## End(Not run)
```

---

peng *Size measurements for penguins near Palmer Station, Antarctica*

Description

Data originally from palmerpenguins. Includes measurements for penguin species, island in Palmer Archipelago, size (flipper length, body mass, bill dimensions), and sex.

Usage

```
peng
```

Format

A tibble with 333 rows and 8 variables:

species a factor denoting penguin species (`"Adélie"`, `"Chinstrap"` or `"Gentoo"`)

island a factor denoting island in Palmer Archipelago, Antarctica (`"Biscoe"`, `"Dream"` or `"Torgersen"`)

bill_length a number denoting bill length (millimeters)

bill_depth a number denoting bill depth (millimeters)

flipper_length an integer denoting flipper length (millimeters)

body_mass an integer denoting body mass (grams)

sex a factor denoting penguin sex (`"f"`, `"m"`)

year an integer denoting the study year (2007, 2008, or 2009)

Details

In this version, variable names have been shortened (removing units) and observations with missing data have been removed.

Source

Adélie penguins: Palmer Station Antarctica LTER and K. Gorman. 2020. Structural size measurements and isotopic signatures of foraging among adult male and female Adélie penguins (Pygoscelis adeliae) nesting along the Palmer Archipelago near Palmer Station, 2007-2009 ver 5. Environmental Data Initiative doi:10.6073/pasta/98b16d7d563f265cb52372c8ca99e60f

Gentoo penguins: Palmer Station Antarctica LTER and K. Gorman. 2020. Structural size measurements and isotopic signatures of foraging among adult male and female Gentoo penguin (Pygoscelis papua) nesting along the Palmer Archipelago near Palmer Station, 2007-2009 ver 5. Environmental Data Initiative doi:10.6073/pasta/7fca67fb28d56ee2ffa3d9370ebda689

Chinstrap penguins: Palmer Station Antarctica LTER and K. Gorman. 2020. Structural size measurements and isotopic signatures of foraging among adult male and female Chinstrap penguin (Pygoscelis antarcticus) nesting along the Palmer Archipelago near Palmer Station, 2007-2009 ver 6. Environmental Data Initiative doi:10.6073/pasta/c14dfcfada8ea13a17536e73eb6fbe9e

Originally published in: Gorman K.B., Williams T.D., Fraser W.R. (2014) Ecological Sexual Dimorphism and Environmental Variability within a Community of Antarctic Penguins (Genus Pygoscelis). PLoS ONE 9(3): e90081. doi:10.1371/journal.pone.0090081

Examples

```
data(peng)
# Covariance ellipses, centered, first two variables
covEllipses(cbind(bill_length, bill_depth) ~ species, data=peng,
     center=TRUE,
     fill=c(rep(FALSE,3), TRUE),
```

```
     fill.alpha=.1, label.pos=c(1:3,0))

# All pairs when more than two variables are specified. They look pretty similar
covEllipses(peng[,3:6], peng$species,
            variables=1:4,
            fill=c(rep(FALSE,3), TRUE),
            fill.alpha=.1)

# Box's M test
peng.boxm <- boxM(cbind(bill_length, bill_depth, flipper_length, body_mass) ~ species, data=peng)
peng.boxm
plot(peng.boxm, gplabel="Species")

# Fit MANOVA model, predicting species
peng.mod0 <-lm(cbind(bill_length, bill_depth, flipper_length, body_mass) ~
               species, data=peng)
car::Anova(peng.mod0)

# HE plot
heplot(peng.mod0, fill=TRUE, fill.alpha=0.1,
       size="effect",
       xlim=c(35,52), ylim=c(14,20))
```

`Plastic` *Plastic Film Data*

Description

An experiment was conducted to determine the optimal conditions for extruding plastic film. Three responses were measured in relation to two factors, change in rate of extrusion and amount of an additive, both with two levels and $n = 5$ observations per cell.

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)
car::Anova(plastic.mod)

pairs(plastic.mod)
```

`plot.boxM` *Plot for Box's M test and generalizations*

Description

This function creates a simple dot chart showing the contributions (log determinants) of the various groups to Box's M test for equality of covariance matrices. An important virtue of these plots is that they can show *how* the groups differ from each other, and from the pooled covariance matrix using a scalar like $ln|S|$. In this way, they can suggest more specific questions or hypotheses regarding the equality of covariance matrices, analogous to the use of contrasts and linear hypotheses for testing differences among group mean vectors.

Because Box's M test is based on a specific function (log determinant) of the covariance matrices in the groups compared to the pooled covariance matrix, this function also also allow plots of other measures based on the eigenvalues of these covariance matrices.

Confidence intervals are only available for the default Box M test, using `which="logDet"`. The theory for this comes from Cai et-al. (2015).

Usage

```
## S3 method for class 'boxM'
plot(
  x,
  gplabel = NULL,
  which = c("logDet", "product", "sum", "precision", "max"),
  log = which == "product",
  pch = c(16, 15),
  cex = c(2, 2.5),
  col = c("blue", "red"),
  rev = FALSE,
  xlim,
  conf = 0.95,
  method = 1,
  bias.adj = TRUE,
```

```
  lwd = 2,
  ...
)
```

Arguments

x	A "boxM" object resulting from boxM()
gplabel	character string used to label the group factor.
which	Measure to be plotted. The default, "logDet", is the standard plot. Other values are: "product", "sum", "precision" and "max"
log	logical; if TRUE, the log of the measure is plotted. The default, which=="product", produces a plot equivalent to the plot of "logDet".
pch	a vector of two point symbols to use for the individual groups and the pooled data, respectively
cex	character size of point symbols, a vector of length two for groups and pooled data, respectively
col	colors for point symbols, a vector of length two for the groups and the pooled data
rev	logical; if TRUE, the order of the groups is reversed on the vertical axis.
xlim	x limits for the plot
conf	coverage for approximate confidence intervals, 0 <= conf < 1 ; use conf=0 to suppress these
method	confidence interval method; see logdetCI()
bias.adj	confidence interval bias adjustment; see logdetCI()
lwd	line width for confidence interval
...	Arguments passed down to graphics::dotchart().

Author(s)

Michael Friendly

References

Cai, T. T., Liang, T., & Zhou, H. H. (2015). Law of log determinant of sample covariance matrix and optimal estimation of differential entropy for high-dimensional Gaussian distributions. *Journal of Multivariate Analysis*, **137**, 161–172. doi:10.1016/j.jmva.2015.02.003.

Friendly, M., & Sigal, M. (2018). Visualizing Tests for Equality of Covariance Matrices. *The American Statistician*, **72**(4); doi:10.1080/00031305.2018.1497537. Online: https://www.datavis.ca/papers/EqCov-TAS.pdf.

See Also

boxM(), logdetCI()

graphics::dotchart()

Other diagnostic plots: cqplot(), distancePlot(), eigstatCI()

Examples

```
# Iris data
res <- boxM(iris[, 1:4], iris[, "Species"])
plot(res, gplabel="Species")

# Skulls data
skulls.mod <- lm(cbind(mb, bh, bl, nh) ~ epoch, data=Skulls)
skulls.boxm <- boxM(skulls.mod)
plot(skulls.boxm, gplabel="Epoch")
plot(skulls.boxm, gplabel="Epoch", bias.adj=FALSE)

# other measures
 plot(skulls.boxm, which="product", gplabel="Epoch", xlim=c(10,14))
 plot(skulls.boxm, which="sum", gplabel="Epoch")
 plot(skulls.boxm, which="precision", gplabel="Epoch")
 plot(skulls.boxm, which="max", gplabel="Epoch")
```

`plot.robmlm` *Plot observation weights from a robust multivariate linear models*

Description

Creates an index plot of the observation weights assigned in the last iteration of `robmlm()`. Observations with low weights have large residual squared distances and are potential multivariate outliers with respect to the fitted model.

Usage

```
## S3 method for class 'robmlm'
plot(
  x,
  labels,
  groups,
  group.axis = TRUE,
  id.weight = 0.7,
  id.pos = 4,
  pch = 19,
  col = palette()[1],
  cex = par("cex"),
  segments = FALSE,
  xlab = "Case index",
  ylab = "Weight in robust MLM",
  ...
)
```

Arguments

x	A "robmlm" object
labels	Observation labels for point identification. If not specified, uses rownames from the original data
groups	Optional grouping variable, a factor with length equal to the number of observations, used to identify groups in the plot.
group.axis	Logical; whether to draw an axis at the top identifying the groups. Not drawn if groups is missing.
id.weight	Threshold for identifying observations with small weights
id.pos	Position of observation label relative to the point
pch	Point symbol(s); can be a vector of length equal to the number of observations in the data frame.
col	Point color(s). Multiple colors can be specified so that each point can be given its own color. If there are fewer colors than points they are recycled in the standard fashion.
cex	Point character size(s)
segments	logical; if TRUE, draw line segments from 1.0 down to the point
xlab	x axis label
ylab	y axis label
...	other arguments passed to graphics::plot()

Value

Returns invisibly the weights for the observations labeled in the plot

Author(s)

Michael Friendly

See Also

robmlm()

Other robust methods: Mahalanobis(), robmlm()

Examples

```
data(Skulls)
sk.rmod <- robmlm(cbind(mb, bh, bl, nh) ~ epoch, data=Skulls)
plot(sk.rmod, col=Skulls$epoch, segments=TRUE)
axis(side=3, at=15+seq(0,120,30), labels=levels(Skulls$epoch), cex.axis=1)

# Pottery data

data(Pottery, package = "carData")
pottery.rmod <- robmlm(cbind(Al,Fe,Mg,Ca,Na)~Site, data=Pottery)
plot(pottery.rmod, col=Pottery$Site, segments=TRUE)
```

```
# SocialCog data

data(SocialCog)
SC.rmod <- robmlm(cbind( MgeEmotions, ToM, ExtBias, PersBias) ~ Dx,
               data=SocialCog)
plot(SC.rmod,
     col=SocialCog$Dx, segments=TRUE)
# label the groups
ctr <- split(seq(nrow(SocialCog)), SocialCog$Dx) |> lapply(mean)
axis(side = 3, at=ctr, labels = names(ctr), cex.axis=1.2)

# use the groups arg
colors = c("red", "darkgreen", "blue")
ids <- plot(SC.rmod,
     groups=SocialCog$Dx,
     col = colors,
     pch = 15:17,
     segments=TRUE)

# the cases labeled and their weights
ids
```

---

plot_boxM_boot *Plot for Box's M test with bootstrap CIs*

Description

Enhanced version of `plot.boxM()` that supports bootstrap confidence intervals for eigenvalue-based statistics in addition to the existing analytic CIs for log determinants.

Usage

```
plot_boxM_boot(
  x,
  Y = NULL,
  group = NULL,
  gplabel = NULL,
  which = c("logDet", "product", "sum", "precision", "max"),
  log = which == "product",
  pch = c(16, 15),
  cex = c(2, 2.5),
  col = c("blue", "red"),
  rev = FALSE,
  xlim,
  conf = 0.95,
  method = 1,
```

```
  bias.adj = TRUE,
  lwd = 2,
  boot.R = 1000,
  boot.type = c("perc", "bca", "norm", "basic"),
  boot.parallel = FALSE,
  boot.ncpus = 2,
  boot.seed = NULL,
  ...
)
```

Arguments

x	A "boxM" object resulting from boxM()
Y	Optional data matrix (required for bootstrap CIs with eigenvalue stats)
group	Optional grouping variable (required for bootstrap CIs with eigenvalue stats)
gplabel	Character string used to label the group factor
which	Measure to be plotted
log	Logical; if TRUE, the log of the measure is plotted
pch	Point symbols for groups and pooled data
cex	Character size of point symbols
col	Colors for point symbols
rev	Logical; if TRUE, reverse order of groups on vertical axis
xlim	X limits for the plot
conf	Coverage for confidence intervals (0 to suppress)
method	CI method for logDet (see logdetCI())
bias.adj	Bias adjustment for logDet CIs
lwd	Line width for confidence intervals
boot.R	Number of bootstrap replicates (for eigenvalue stats only)
boot.type	Type of bootstrap CI ("perc", "bca", "norm", "basic")
boot.parallel	Use parallel processing for bootstrap
boot.ncpus	Number of CPUs for parallel bootstrap
boot.seed	Random seed for bootstrap reproducibility
...	Additional arguments passed to dotchart()

Details

This implementation is still **Experimental**

Value

Invisibly returns the confidence interval data frame (if computed)

Examples

```
## Not run:
library(boot)
# source("dev/eigstatCI.R")
# source("dev/plot.boxM_with_bootstrap.R")

# Iris data with bootstrap CIs
boxm <- boxM(iris[,1:4], iris$Species)

# logDet with analytic CI (same as before)
plot_boxM_boot(boxm, gplabel = "Species")

# Sum of eigenvalues with bootstrap CI
plot_boxM_boot(boxm, Y = iris[,1:4], group = iris$Species,
               which = "sum", gplabel = "Species", boot.R = 1000)

## End(Not run)
```

`Pottery2` *Chemical Analysis of Romano-British Pottery*

Description

Results of chemical analyses of 48 specimens of Romano-British pottery published by Tubb et al. (1980). The numbers are the percentage of various metal oxides found in each sample for elements of concentrations greater than 0.01\ contrast to `carData::Pottery`.

Format

A data frame with 48 observations on the following 12 variables.

`Region` a factor with levels `Gl` `NF` `Wales`

`Site` a factor with levels `AshleyRails` `Caldicot` `Gloucester` `IsleThorns` `Llanedryn`

`Kiln` a factor with levels `1` `2` `3` `4` `5`

`Al` amount of aluminum oxide, Al_2O_3

`Fe` amount of iron oxide, Fe_2O_3

`Mg` amount of magnesium oxide, MgO

`Ca` amount of calcium oxide, CaO

`Na` amount of sodium oxide, Na_2O

`K` amount of potassium oxide, K_2O

`Ti` amount of titanium oxide, TiO_2

`Mn` amount of manganese oxide, MnO

`Ba` amount of BaO

Details

The specimens are identified by their `rownames` in the data frame. `Kiln` indicates at which kiln site the pottery was found; `Site` gives the location names of those sites. The kiln sites come from three `Regions`, `("Gl"=1, "Wales"=(2, 3), "NF"=(4, 5))`, where the full names are "Gloucester", "Wales", and "New Forrest".

The variable `Kiln` comes pre-supplied with contrasts to test interesting hypotheses related to `Site` and `Region`.

Source

Originally slightly modified from files by David Carlson, now at `archdata::RBPottery`. %

References

Baxter, M. J. 2003. *Statistics in Archaeology*. Arnold, London.

Carlson, David L. 2017. *Quantitative Methods in Archaeology Using R*. Cambridge University Press, pp 247-255, 335-342.

Tubb, A., A. J. Parker, and G. Nickless. 1980. The Analysis of Romano-British Pottery by Atomic Absorption Spectrophotometry. *Archaeometry*, **22**, 153-171.

See Also

`carData::Pottery` for the related (subset) data set; `archdata::RBPottery` for a newer version with more variables.

Examples

```
library(car)
data(Pottery2)
# contrasts for Kiln correspond to between Region [,1:2] and within Region [,3:4]
contrasts(Pottery2$Kiln)

pmod <-lm(cbind(Al,Fe,Mg,Ca,Na,K,Ti,Mn,Ba)~Kiln, data=Pottery2)
car::Anova(pmod)

# extract coefficient names for linearHypotheses
coefs <- rownames(coef(pmod))[-1]

# test differences among regions
linearHypothesis(pmod, coefs[1:2])
# test differences within regions B, C
linearHypothesis(pmod, coefs[3:4])

heplot(pmod, fill=c(TRUE,FALSE), hypotheses=list("Region" =coefs[1:2], "WithinBC"=coefs[3:4]))

# all pairwise views;  note that Ba shows no effect
pairs(pmod, fill=c(TRUE,FALSE))

# canonical view, via candisc::heplot
```

```
if (require(candisc)) {

# canonical analysis: how many dimensions?
(pcan <- candisc(pmod))

heplot(pcan, scale=18, fill=c(TRUE,FALSE), var.col="darkgreen", var.lwd=2, var.cex=1.5)

## Not run:
heplot3d(pcan, scale=8)

## End(Not run)
}
```

`Probe` *Response Speed in a Probe Experiment*

Description

Data from a probe experiment testing whether immediate memory for sentences is influenced by the phrase structure of the sentence. The data sets come from Timm (1975), Ex. 3.14 and Ex. 3.16 (p.244)

Format

`Probe1`: A data frame with 11 observations on the following 5 variables.

`p1` speed at position 1

`p2` speed at position 2

`p3` speed at position 3

`p4` speed at position 4

`p5` speed at position 5

`Probe2`: A data frame with 20 observations on the following 6 variables.

`stm` Short term memory capacity: a factor with levels `High` `Low`

`p1` speed at position 1

`p2` speed at position 2

`p3` speed at position 3

`p4` speed at position 4

`p5` speed at position 5

Details

Procedure: Subjects listened to tape-recorded sentences. Each sentence was followed by a "probe word" from one of 5 positions within the sentence. The subject had to respond with the word which immediately followed the probe word in the sentence. The dependent measure is response speed = k(1/reaction time).

Sample sentence:

```
*       The tall man met the young girl who got the new hat.
Pos'ns:      1    2            3    4    5
Function:   ADJ1 SUBJ        ADJ2  OBJ  REL.PN
```

In Probe2, there are two groups of subjects, pre-selected on a test of short term memory.

These data sets (fictitious) are used as examples of single-sample and two-sample profile analysis or simple repeated measure designs with structured contrasts.

Source

Timm, N. (1975) *Multivariate analysis, with applications in education and psychology* Brooks/Cole.

Examples

```
data(Probe1)
boxplot(Probe1)

pmod1 <- lm(cbind(p1,p2,p3,p4,p5) ~ 1, data=Probe1)
idata <- data.frame(position=factor(1:5))

library(car)
(pmod1.aov <- car::Anova(pmod1, idata=idata, idesign=~position))

# using default contrasts (p5 as reference level)
heplot(pmod1, manova=pmod1.aov,
       iterm="position",
       type="III",
       idata=idata, idesign=~position)
pairs(pmod1, manova=pmod1.aov,
       iterm="position",
       type="III",
       idata=idata, idesign=~position)

# contrasts for substantative hypotheses regarding
# sentence position effects
C <- matrix(c(
 1, 1, -1, -1, 0,
 1, -1, 1, -1, 0,
 1, -1, -1, 1, 0,
 1, 1, 1, 1, -4), 5, 4)
rownames(C) <- paste("p", 1:5, sep="")
colnames(C) <- c("SubPred", "AdjNoun", "SPxAN", "RelPN")
```

```
contrasts(idata$position)<- C
(pmod1.aov <- car::Anova(pmod1, idata=idata, idesign=~position))

heplot(pmod1, manova=pmod1.aov,
       iterm="position", type="III", idata=idata, idesign=~position)
pairs(pmod1, manova=pmod1.aov,
       iterm="position", type="III", idata=idata, idesign=~position)
```

---

pvPlot *Partial Variables Plot*

Description

A partial variable plot is a visualization of a partial correlation of two variables in the context of other variables in a dataset. For two variables x_i and x_j, it is simply an enhanced scatterplot of the *partial residuals*, $e_i = (x_i - \hat{x}_i)$ from a regression of x_i on all other variables Z against those $e_j = (x_j - \hat{x}_j)$ for another variable x_j. Consequently, it shows directly the net, *conditional relation* between $x_i, x_j|$others when all of the others in Z have been controlled/adjusted-for.

As implemented here, the basic scatterplot of these residuals can be enhanced by also showing the data ellipse of these residuals, the linear regression line, which reflects the partial correlation, and point labels to identify unusual data.

Usage

```
pvPlot(
  X,
  vars = 1:2,
  others = NULL,
  labels,
  id = FALSE,
  ellipse = TRUE,
 ellipse.args = list(levels = 0.68, fill = TRUE, fill.alpha = 0.05, robust = FALSE, col
    = "black"),
  draw = TRUE,
  col = "black",
  pch = 16,
  cex = par("cex"),
  axes = TRUE,
  regline = TRUE,
  show.partial = list(loc = c(0.025, 0.95), cex = 1.2),
  ...
)
```

Arguments

X	a data.frame (or tibble) of numeric variables
vars	either the character names of two variables in X or their indices
others	character names or indices of the variables to partial out. If NULL (the default), all variables in X other than vars are used.
labels	id labels for the points. If not supplied, rownames of the dataset are used.
id	controls point identification; if FALSE (the default), no points are identified; can be a list of named arguments to the car::showLabels() function
ellipse	logical; whether to draw the data ellipse
ellipse.args	a list of arguments controlling the ellipse: levels, fill, fill.alpha, robust, and col (ellipse outline/fill color, independent of the point col). See car::dataEllipse() for what these mean.
draw	logical; if TRUE produce graphical output; if FALSE, only invisibly return coordinates of ellipse(s).
col	color used for points
pch	the plotting character for points
cex	Character expansion for points and labels
axes	logical; if TRUE (the default), grey axes lines are drawn at 0 on both coordinates
regline	controls the regression line. FALSE suppresses it; TRUE (default) draws it with default style; a list with named elements col and/or lwd draws it with those attributes, e.g. regline = list(col="red", lwd=3).
show.partial	controls whether the partial correlation value is displayed in the plot. If FALSE the value is not shown. Otherwise, can be a list containing the location (loc) and character size (cex) of the label.
...	other arguments passed to car::dataEllipse()

Details

Partial variable plots are intimately related to an *added-variable plot*, such as produced by car::avPlots(). However, that implementation is designed for a linear model, rather than a data.frame.

The present version assumes that all variables passed are numeric.

This function uses car::dataEllipse() for drawing, so further documentation of arguments passed there should be consulted.

Value

```
This functions is mainly used for their side effect of producing plots. For greater
     flexibility (e.g., adding plot annotations), it returns invisibly the coordinates of the residual
        plotted.
```

Author(s)

Michael Friendly

Examples

```
data(crime, package = "ggbiplot")
crime.num <- crime |>
  tibble::column_to_rownames("st") |>
  dplyr::select(where(is.numeric))

pvPlot(crime.num, vars = c("burglary", "larceny"))
pvPlot(crime.num, vars = c("auto", "robbery"))

# ellipse color independent of point color
pvPlot(crime.num, vars = c("auto", "robbery"),
       ellipse.args = list(col="red"))

# styled regression line
pvPlot(crime.num, vars = c("burglary", "larceny"),
       regline = list(col = "red", lwd = 3))

# partial out only a subset of the other variables
pvPlot(crime.num, vars = c("burglary", "larceny"),
       others = c("murder", "rape"))

# suppress the ellipse
pvPlot(crime.num, vars = c("burglary", "larceny"), ellipse=FALSE)

# label some observations
pvPlot(crime.num, vars = c("burglary", "larceny"),
       id = list(n=5),
       cex.lab = 1.5)
```

`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.

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. https://rdrr.io/rforge/ALA/.

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. doi:10.18637/jss.v037.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))
car::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.

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*, https://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. doi:10.18637/jss.v037.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"))
car::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)
 }
 )
```

`ReadingDisability` *Cognitive and Achievement Test Scores by Reading Level*

Description

Six cognitive/achievement test scores for children classified into four reading-achievement groups (`Severe`, `Mild`, `Average`, `Superior`), reconstructed from the group means, standard deviations, and pooled within-cells correlations reported as a worked "real data example" in Bray & Maxwell (1985), Table 2.3.

Individual rows are **simulated**, not the original observations: only group-level summary statistics were published, with no raw data and no per-group correlations (a single correlation matrix, pooled across all four groups, was reported). Each group was drawn from a multivariate normal with that group's reported means and a covariance matrix built from that group's reported SDs combined with the pooled correlations, then forced to match those exact moments via `MASS::mvrnorm(..., empirical = TRUE)`. Summarizing this data frame by `Group` exactly reproduces Bray & Maxwell's Table 2.3 (see `data-raw/ReadingDisability.R`).

B&M (p.38) describe the six measures as follows: "The PPVT is a general verbal measure of IQ, whereas the VF and SIM tests are verbal criteria. The EF taps higher-order nonverbal abilities, whereas the VMI and RD are general nonverbal measures."

Usage

```
data("ReadingDisability")
```

Format

A data frame with 571 observations on the following 7 variables, a 4-level between-subjects factor (`Group`) with unequal n.

`Group` Reading achievement level, an ordered factor with levels (worst to best) `Severe` (n = 93) < `Mild` (n = 113) < `Average` (n = 274) < `Superior` (n = 91)

`PPVT` Peabody Picture Vocabulary Test score – a general verbal measure of IQ, numeric

`RD` Recognition-Discrimination Test score – a general nonverbal measure, numeric

`EF` Embedded Figures Test score – taps higher-order nonverbal abilities, numeric

`VF` Verbal Fluency Test score – a verbal criterion measure, numeric

`VMI` Beery Visual-Motor Integration Test score – a general nonverbal measure, numeric

`SIM` Similarities subtest of the Wechsler Preschool and Primary Scale of Intelligence (WPPSI) score – a verbal criterion measure, numeric

Source

Bray, J. H., & Maxwell, S. E. (1985). *Multivariate Analysis of Variance*. Sage. Table 2.3, p.38, "Means, Standard Deviations, and Within-Cells Correlations for Real Data Example."

B&M attribute this table to Fletcher, J. M., & Satz, P. (1980), but **that reference does not appear in B&M's own References section** – a gap in the source book, not (so far as we can tell) a typo on our part. A copy of Fletcher, J. M., & Satz, P. (1980), *Developmental changes in the neuropsychological correlates of reading achievement: A six-year longitudinal follow-up*, Journal of Clinical Neuropsychology, 2(1), 23-37, is available, but its own reported variables differ from the six tabulated here – so it is not confirmed to be the exact source of Table 2.3, only the most likely candidate (same authors, matching year, matching four-group reading-disability design). Treat the B&M attribution as unresolved pending a source that actually contains these six measures.

Examples

```
data(ReadingDisability)
str(ReadingDisability)

rd.mod <- lm(cbind(PPVT, RD, EF, VF, VMI, SIM) ~ Group,
             data = ReadingDisability)
car::Anova(rd.mod)

heplot(rd.mod, fill = TRUE, fill.alpha = 0.1)
```

---

rel_diff *Relative Difference between two Arrays or Data Frames*

Description

Calculates the relative difference, defined as

$$\frac{|x - y|}{x}$$

between two arrays or data frames, where `x` are considered reference values.

Usage

```
rel_diff(x, y, pct = TRUE, epsilon = 10 * .Machine$double.eps)
```

Arguments

x	An array or data frame, considered the reference values
y	Comparison array or data frame
pct	Logical; if TRUE the relative differences are multiplied by 100, giving values in percent difference from x.
epsilon	Threshold for values near zero

Details

Beyond the obvious, a natural use case is to compare coefficients for alternative models for the same data, e.g., a classical and a robust model.

Value

An array or data frame the same size as x and y containing the relative differences

See Also

robmlm()

Examples

```
# simple example
m1 <- cbind(c(0,1), c(1,1))
m2 <- cbind(c(0,1), c(1.01,1.11))
rel_diff(m1, m2, pct = FALSE)
rel_diff(m1, m2)

# compare coefficients
data(Skulls)

# fit manova model, classically and robustly
sk.mlm <- lm(cbind(mb, bh, bl, nh) ~ epoch, data=Skulls)
sk.rlm <- robmlm(cbind(mb, bh, bl, nh) ~ epoch, data=Skulls)
rel_diff(coef(sk.mlm),
         coef(sk.rlm))
```

robmlm *Robust Fitting of Multivariate Linear Models*

Description

Fit a multivariate linear model by robust regression using a simple M estimator that down-weights observations with large residuals

Fitting is done by iterated re-weighted least squares (IWLS), using weights based on the Mahalanobis squared distances of the current residuals from the origin, and a scaling (covariance) matrix calculated by `MASS::cov.trob()`. The design of these methods were loosely modeled on `MASS::rlm()`.

These S3 methods are designed to provide a specification of a class of robust methods which extend `mlm`s, and are therefore compatible with other `mlm` extensions, including `car::Anova()` and `heplot()`.

An internal `vcov.mlm` function is an extension of the standard `stats::vcov()` method providing for the use of observation weights. A `plot.robmlm()` method provides simple index plots of case weights to visualize those that were down-weighted.

Usage

```
robmlm(X, ...)

## Default S3 method:
robmlm(
  X,
  Y,
  w,
  P = 2 * pnorm(4.685, lower.tail = FALSE),
  tune,
  max.iter = 100,
  psi = psi.bisquare,
  tol = 1e-06,
  initialize,
  verbose = FALSE,
  ...
)

## S3 method for class 'formula'
robmlm(
  formula,
  data,
  subset,
  weights,
  na.action,
  model = TRUE,
  contrasts = NULL,
```

```
  ...
)

## S3 method for class 'mlm'
robmlm(X, ...)

## S3 method for class 'robmlm'
print(x, ...)

## S3 method for class 'robmlm'
summary(object, ...)

## S3 method for class 'summary.robmlm'
print(x, ...)
```

Arguments

X	for the default method, a model matrix, including the constant (if present); for the mlm method, a fitted classical mlm object (e.g., from lm(cbind(y1, y2, ...) ~ ...)), which is refit robustly using the same response, predictors, contrasts, and any subset, weights or na.action given in the original lm() call. These can't be changed in the call to robmlm() (an error is signaled); only the control arguments of the default method (tune, max.iter, ...) can be given. The result records the call as if it had been fit with the formula method, so stats::update() works on it.
...	other arguments, passed down. In particular relevant control arguments can be passed to the to the robmlm.default method.
Y	for the default method, a response matrix
w	prior observation weights
P	two-tail probability, to find cutoff quantile for chisq (tuning constant); default is set for bisquare weight function
tune	tuning constant (if given directly)
max.iter	maximum number of iterations
psi	robustness weight function; MASS::psi.bisquare() is the default
tol	convergence tolerance, maximum relative change in coefficients
initialize	modeling function to find start values for coefficients, equation-by-equation; if absent WLS (lm.wfit) is used
verbose	show iteration history? (TRUE or FALSE)
formula	a formula of the form cbind(y1, y2, ...) ~ x1 + x2 + ...
data	a data frame from which variables specified in formula are preferentially to be taken.
subset	An index vector specifying the cases to be used in fitting.
weights	a vector of prior weights for each case.
na.action	A function to specify the action to be taken if NAs are found. The 'factory-fresh' default action in R is stats::na.omit(), and can be changed by base::options(na.action=).

`model`	should the model frame be returned in the object?
`contrasts`	optional contrast specifications; see `stats::lm()` for details.
`x`	a `robmlm` object
`object`	a `robmlm` object

Details

Weighted least squares provides a method for correcting a variety of problems in linear models by estimating parameters that minimize the *weighted* sum of squares of residuals $\Sigma w_i e_i^2$ for specified weights $w_i, i = 1, 2, \ldots n$.

M-estimation generalizes this by minimizing the sum of a symmetric function $\rho(e_i)$ of the residuals, where the function is designed to reduce the influence of outliers or badly fit observations. The function $\rho(e_i) = |e_i|$ minimizes the least absolute values, while the *bisquare* function uses an upper bound on influence. For multivariate problems, a simple method is to use Mahalanobis $D^2(\mathbf{e}_i)$ to calculate the weights.

Because the weights and the estimated coefficients depend on each other, this is done iteratively, computing weights and then re-estimating the model with those weights until convergence.

Value

An object of class `"robmlm"` inheriting from `c("mlm", "lm")`.

This means that the returned `"robmlm"` contains all the components of `"mlm"` objects described for `stats::lm()`, plus the following:

weights final observation weights

iterations number of iterations

converged logical: did the IWLS process converge?

The generic accessor functions `stats::coefficients()`, `stats::effects()`, `stats::fitted.values()` and `stats::residuals()` extract various useful features of the value returned by `robmlm`.

Author(s)

John Fox; packaged by Michael Friendly

References

A. Marazzi (1993) *Algorithms, Routines and S Functions for Robust Statistics*. Wadsworth & Brooks/Cole.

See Also

`plot.robmlm()` for a plot method; `MASS::rlm()`, `MASS::cov.trob()`

Other robust methods: `Mahalanobis()`, `plot.robmlm()`

Examples

```
# Skulls data
# -----------
data(Skulls)

# make shorter labels for epochs and nicer variable labels in heplots
Skulls$epoch <- factor(Skulls$epoch, labels=sub("c","",levels(Skulls$epoch)))
# variable labels
vlab <- c("maxBreadth", "basibHeight", "basialLength", "nasalHeight")

# fit manova model, classically and robustly
sk.mod <- lm(cbind(mb, bh, bl, nh) ~ epoch, data=Skulls)
sk.rmod <- robmlm(cbind(mb, bh, bl, nh) ~ epoch, data=Skulls)

# equivalently, refit an existing classical mlm object directly
sk.rmod2 <- robmlm(sk.mod)
all.equal(coef(sk.rmod), coef(sk.rmod2))
# the result can be updated like one fit from a formula
update(sk.rmod2, subset = epoch != "200BC")

# standard mlm methods apply here
coefficients(sk.rmod)

# index plot of weights
plot(sk.rmod, segments = TRUE, col = Skulls$epoch)
points(sk.rmod$weights, pch=16, col=Skulls$epoch)
text(x = 15+seq(0,120,30), y = 1.05, labels=levels(Skulls$epoch), xpd=TRUE)

# heplots to see effect of robmlm vs. mlm
heplot(sk.mod, hypotheses=list(Lin="epoch.L", Quad="epoch.Q"),
    xlab=vlab[1], ylab=vlab[2], cex=1.25, lty=1)
heplot(sk.rmod, hypotheses=list(Lin="epoch.L", Quad="epoch.Q"),
    add=TRUE, error.ellipse=TRUE, lwd=c(2,2), lty=c(2,2),
    term.labels=FALSE, hyp.labels=FALSE, err.label="")

###############
# Pottery data

data(Pottery, package = "carData")
pottery.mod <- lm(cbind(Al,Fe,Mg,Ca,Na)~Site, data=Pottery)
pottery.rmod <- robmlm(cbind(Al,Fe,Mg,Ca,Na)~Site, data=Pottery)
car::Anova(pottery.mod)
car::Anova(pottery.rmod)

# index plot of weights
plot(pottery.rmod$weights, type="h")
points(pottery.rmod$weights, pch=16, col=Pottery$Site)

# heplots to see effect of robmlm vs. mlm
heplot(pottery.mod, cex=1.3, lty=1)
heplot(pottery.rmod, add=TRUE, error.ellipse=TRUE, lwd=c(2,2), lty=c(2,2),
    term.labels=FALSE, err.label="")
```

```
###############
# Prestige data
data(Prestige, package = "carData")

# treat women and prestige as response variables for this example
prestige.mod <- lm(cbind(women, prestige) ~ income + education + type, data=Prestige)
prestige.rmod <- robmlm(cbind(women, prestige) ~ income + education + type, data=Prestige)

coef(prestige.mod)
coef(prestige.rmod)
# how much do coefficients change?
round(coef(prestige.mod) - coef(prestige.rmod),3)

# pretty plot of case weights
plot(prestige.rmod$weights, type="h", xlab="Case Index", ylab="Robust mlm weight", col="gray")
points(prestige.rmod$weights, pch=16, col=Prestige$type)
legend(0, 0.7, levels(Prestige$type), pch=16, col=palette()[1:3], bg="white")

heplot(prestige.mod, cex=1.4, lty=1)
heplot(prestige.rmod, add=TRUE, error.ellipse=TRUE, lwd=c(2,2), lty=c(2,2),
    term.labels=FALSE, err.label="")
```

Rohwer *Rohwer Data Set*

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.

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 tasks, n, s, ns, na, and ss, which differed in the syntactic and semantic relationship between the stimulus and response words in each pair.

Timm (1975) does not give a source, but the most relevant studies are Rohwer & Levin (1968) and Rohwer & Ammons (1971). The paired-associate tasks are described as:

n (named): Simple paired-associate task where participants learn pairs of nouns with no additional context

s (sentence): Participants learn pairs embedded within a sentence

ns (named sentence): A combination where participants learn noun pairs with sentence context

na (named action): Pairs are learned with an action relationship between them

ss (sentence still): Similar to the sentence condition but with static presentation

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

Rohwer, W.D., Jr., & Levin, J.R. (1968). Action, meaning and stimulus selection in paired-associate learning. *Journal of Verbal Learning and Verbal Behavior*, **7**: 137-141.

Rohwer, W. D., Jr., & Ammons, M. S. (1971). Elaboration training and paired-associate learning efficiency in children. *Journal of Educational Psychology*, **62**(5), 376-383.

Examples

```
str(Rohwer)

# Plot responses against each predictor
library(tidyr)
library(dplyr)
library(ggplot2)

yvars <- c("SAT", "PPVT", "Raven" )
xvars <- c("n", "s", "ns", "na", "ss")
Rohwer_long <- Rohwer %>%
  pivot_longer(cols = all_of(xvars), names_to = "xvar", values_to = "x") |>
  pivot_longer(cols = all_of(yvars), names_to = "yvar", values_to = "y") |>
  mutate(xvar = factor(xvar, xvars), yvar = factor(yvar, yvars))

ggplot(Rohwer_long, aes(x, y, color = SES, shape = SES, fill = SES)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, formula = y ~ x) +
  stat_ellipse(geom = "polygon", level = 0.68, alpha = 0.1) +
```

```
  facet_grid(yvar ~ xvar, scales = "free") +
  labs(x = "predictor", y = "response") +
  theme_bw(base_size = 14)


## ANCOVA, assuming equal slopes
rohwer.mod <- lm(cbind(SAT, PPVT, Raven) ~ SES + n + s + ns + na + ss, data=Rohwer)
car::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
## Not run:
col <- c("red", "green3", "blue", "cyan", "magenta", "brown", "gray")
heplot3d(rohwer.mod, hypotheses=list("Regr" = c("n", "s", "ns", "na", "ss")),
                     col=col, wire=FALSE)

## End(Not run)

## 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. How do the measures of size vary with the type of 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

```
library(car)
data(RootStock)
str(RootStock)

root.mod <- lm(cbind(girth4, ext4, girth15, weight15) ~ rootstock, data=RootStock)
car::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)
car::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,]))
```

Sake *Taste Ratings of Japanese Rice Wine (Sake)*

Description

Siotani et al. (1985) describe a study of Japanese rice wine (sake) used to investigate the relationship between two subjective ratings (`taste` and `smell`) and a number of physical measurements on 30 brands of sake.

Format

A data frame with 30 observations on the following 10 variables.

`taste` mean taste rating

`smell` mean smell rating

`pH` pH measurement

`acidity1` one measure of acidity

`acidity2` another measure of acidity

`sake` Sake-meter score

`rsugar` direct reducing sugar content

`tsugar` total sugar content

`alcohol` alcohol content

`nitrogen` formol-nitrogen content

Details

These data provide one example of a case where a multivariate regression doesn't benefit from having multiple outcome measures, using the standard tests. Barrett (2003) uses this data to illustrate influence measures for multivariate regression models.

The `taste` and `smell` values are the mean ratings of 10 experts on some unknown scale.

Source

Siotani, M. Hayakawa, T. & Fujikoshi, Y. (1985). *Modern Multivariate Statistical Analysis: A Graduate Course and Handbook*. American Sciences Press, p. 217.

References

Barrett, B. E. (2003). Understanding Influence in Multivariate Regression. *Communications in Statistics - Theory and Methods* 32 (3), 667-680.

Examples

```
data(Sake)
# quick look at the data
boxplot(scale(Sake))

Sake.mod <- lm(cbind(taste,smell) ~ ., data=Sake)

library(car)
car::Anova(Sake.mod)

predictors <- colnames(Sake)[-(1:2)]
# overall multivariate regression test
linearHypothesis(Sake.mod, predictors)

heplot(Sake.mod, hypotheses=list("Regr" = predictors))
```

schooldata *School Data*

Description

School Data, from Charnes et al. (1981), a large scale social experiment in public school education. It was conceived in the late 1960's as a federally sponsored program charged with providing remedial assistance to educationally disadvantaged early primary school students. One aim is to explain scores on 3 different tests, `reading`, `mathematics` and `selfesteem` from 70 school sites by means of 5 explanatory variables related to parents and teachers.

Format

A data frame with 70 observations on the following 8 variables.

`education` Education level of mother as measured by the percentage of high school graduates among female parents

`occupation` Highest occupation of a family member according to a pre-arranged rating scale

`visit` Parental visits index, representing the number of visits to the school site

`counseling` Parent counseling index, calculated from data on time spent with child on school-related topics such as reading together, etc.

`teacher` Number of teachers at the given site

`reading` Reading score as measured by the Metropolitan Achievement Test

`mathematics` Mathematics score as measured by the Metropolitan Achievement Test

`selfesteem` Coopersmith Self-Esteem Inventory, intended as a measure of self-esteem

Details

A number of observations are unusual, a fact only revealed by plotting.

The study was designed to compare schools using Program Follow Through (PFT) management methods of taking actions to achieve goals with those of Non Follow Through (NFT). Observations `1:49` came from PFT sites and `50:70` from NFT sites. This and other descriptors are contained in the dataset `schoolsites`.

Source

This dataset was came originally from the (now-defunct) `FRB` package.

References

A. Charnes, W.W. Cooper and E. Rhodes (1981). Evaluating Program and Managerial Efficiency: An Application of Data Envelopment Analysis to Program Follow Through. *Management Science*, **27**, 668-697.

See Also

schoolsites

Examples

```
data(schooldata)
# initial screening
plot(schooldata)

# better plot
library(corrgram)
corrgram(schooldata,
         lower.panel=panel.ellipse,
         upper.panel=panel.pts)

# check for multivariate outliers
res <- cqplot(schooldata, id.n = 5)
res

#fit the MMreg model
school.mod <- lm(cbind(reading, mathematics, selfesteem) ~
               education + occupation + visit + counseling + teacher, data=schooldata)

# shorthand: fit all others
school.mod <- lm(cbind(reading, mathematics, selfesteem) ~ ., data=schooldata)
car::Anova(school.mod)

# HE plots
heplot(school.mod, fill=TRUE, fill.alpha=0.1)
pairs(school.mod, fill=TRUE, fill.alpha=0.1)

# robust model, using robmlm()
school.rmod <- robmlm(cbind(reading, mathematics, selfesteem) ~ ., data=schooldata)
# note that counseling is now significant
car::Anova(school.rmod)

# Index plot of the weights
wts <- school.rmod$weights
notable <- which(wts < 0.8)
plot(wts, type = "h", col="gray", ylab = "Observation weight")
points(seq_along(wts), wts,
       pch=16,
       col = ifelse(wts < 0.8, "red", "black"))

text(notable, wts[notable],
     labels = notable,
     pos = 3,
     col = "red")



# compare classical HE plot with that based on the robust model
```

```
heplot(school.mod, cex=1.4, lty=1, fill=TRUE, fill.alpha=0.1)
heplot(school.rmod,
       add=TRUE,
       error.ellipse=TRUE,
       lwd=c(2,2), lty=c(2,2),
      term.labels=FALSE, err.label="",
      fill=TRUE)
```

`schoolsites` *Schooldata Sites*

Description

Descriptors for the sites of the `schooldata` dataset, from Charnes et al. (1981). The study was designed to compare schools using Program Follow Through (PFT) management methods of taking actions to achieve goals with those of Non Follow Through (NFT). Observations `1:49` came from PFT sites and `50:70` from NFT sites. This dataset gives other descriptors for the sites, from their Exhibit C.

Usage

```
data("schoolsites")
```

Format

A data frame with 70 observations on the following 7 variables.

`site` site number, a numeric vector

`type` program type, a factor with levels `PFT` ("Program Follow Through") and `NFT` ("Non Follow Through")

`model` education style model, a factor with levels `BA`, `Bank Street`, `California Process`, `Cognitive Curriculum`, `DIM`, `EDC`, `Home-School`, `ILM`, `Parent Education`, `Responsive Education`, `SEDL`, `TEEM`

`site_name` location of site, a character vector

`region` US region, a factor with levels `NC`, `NE`, `S`, `W`

`city_size` city size, an ordered factor with levels `Rural` < `Small` < `Medium` < `Large`

`student_pop` size of the student population, a numeric vector

% @details %% ~~ If necessary, more details than the **description** above ~~

Source

A. Charnes, W.W. Cooper and E. Rhodes (1981). Evaluating Program and Managerial Efficiency: An Application of Data Envelopment Analysis to Program Follow Through. *Management Science*, **27**, 668-697, Exhibit C.

See Also

schooldata

Examples

```
data(schoolsites)
str(schoolsites)
schools <- cbind(schooldata, schoolsites)
schools.mod <- lm(cbind(reading, mathematics, selfesteem) ~
                    education + occupation + visit + counseling + teacher +
                    type + region, data = schools)
car::Anova(schools.mod)
```

Skulls *Egyptian Skulls*

Description

Measurements made on Egyptian skulls from five epochs.

Format

A data frame with 150 observations on the following 5 variables.

epoch the epoch the skull as assigned to, an ordered factor with levels c4000BC c3300BC, c1850BC, c200BC, and cAD150, where the years are only given approximately, of course.

mb maximal breadth of the skull.

bh basibregmatic height of the skull.

bl basialiveolar length of the skull.

nh nasal height of the skull.

Details

The epochs correspond to the following periods of Egyptian history:

1. the early predynastic period (circa 4000 BC);
2. the late predynastic period (circa 3300 BC);
3. the 12th and 13th dynasties (circa 1850 BC);
4. the Ptolemiac period (circa 200 BC);
5. the Roman period (circa 150 AD).

The question is whether the measurements change over time. Non-constant measurements of the skulls over time would indicate interbreeding with immigrant populations.

Note that using polynomial contrasts for epoch essentially treats the time points as equally spaced.

Source

D. J. Hand, F. Daly, A. D. Lunn, K. J. McConway and E. Ostrowski (1994). *A Handbook of Small Datasets*, Chapman and Hall/CRC, London.

References

Thomson, A. and Randall-Maciver, R. (1905) *Ancient Races of the Thebaid*, Oxford: Oxford University Press.

Hand, D. J., F. Daly, A. D. Lunn, K. J. McConway and E. Ostrowski (1994). *A Handbook of Small Datasets*, Chapman and Hall/CRC, London.

Examples

```
data(Skulls)
library(car)    # for Anova

# make shorter labels for epochs
Skulls$epoch <- factor(Skulls$epoch, labels=sub("c","",levels(Skulls$epoch)))

# longer variable labels
vlab <- c("maxBreadth", "basibHeight", "basialLength", "nasalHeight")

# fit manova model
sk.mod <- lm(cbind(mb, bh, bl, nh) ~ epoch, data=Skulls)

Anova(sk.mod)
summary(Anova(sk.mod))

# test trends over epochs
print(linearHypothesis(sk.mod, "epoch.L"), SSP=FALSE) # linear component
print(linearHypothesis(sk.mod, "epoch.Q"), SSP=FALSE) # quadratic component

# typical scatterplots are not very informative
scatterplot(mb ~ bh|epoch, data=Skulls,
            ellipse = list(levels=0.68),
            smooth=FALSE,
            legend = list(coords="topright"),
            xlab=vlab[2], ylab=vlab[1])

scatterplot(mb ~ bl|epoch, data=Skulls,
            ellipse = list(levels=0.68),
            smooth=FALSE,
            legend = list(coords="topright"),
            xlab=vlab[3], ylab=vlab[1])

# HE plots

heplot(sk.mod,
       hypotheses=list(Lin="epoch.L", Quad="epoch.Q"),
       xlab=vlab[1], ylab=vlab[2])
```

```
pairs(sk.mod,
      hypotheses=list(Lin="epoch.L", Quad="epoch.Q"),
      var.labels=vlab)

# 3D plot shows that nearly all of hypothesis variation is linear!
## Not run:
heplot3d(sk.mod, hypotheses=list(Lin="epoch.L", Quad="epoch.Q"), col=c("pink", "blue"))

# view in canonical space
if (require(candisc)) {
 sk.can <- candisc(sk.mod)
 sk.can
 heplot(sk.can)
 heplot3d(sk.can)
}

## End(Not run)
```

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.

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

midterm1, midterm2, final, and possibly eval are the response variables. All other variables are potential predictors.

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)

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

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

## Not run:
heplot3d(grades.mod, col=clr, wire=FALSE)

## End(Not run)

if (require(candisc)) {
 # calculate canonical results for all terms
 grades.can <- candiscList(grades.mod)
 # extract canonical R^2s
 unlist(lapply(grades.can, function(x) x$canrsq))
 # plot class effect in canonical space
 heplot(grades.can, term="class", scale=4)

 # 1 df terms: show canonical scores and weights for responses
 plot(grades.can, term="sex")
 plot(grades.can, term="gpa")
 plot(grades.can, term="boards")
 }
```

SocialCog *Social Cognitive Measures in Psychiatric Groups*

Description

The general purpose of the study (Hartman, 2016, Heinrichs et al. (2015)) was to evaluate patterns and levels of performance on neurocognitive measures among individuals with schizophrenia and schizoaffective disorder using a well-validated, comprehensive neurocognitive battery specifically designed for individuals with psychosis (Heinrichs et al. (2008))

Format

A data frame with 139 observations on the following 5 variables.

`Dx` Diagnostic group, a factor with levels `Schizophrenia`, `Schizoaffective`, `Control`

`MgeEmotions` Score on the Managing emotions test, a numeric vector

`ToM` Score on the The Reading the Mind in the Eyes test (theory of mind), a numeric vector

`ExtBias` Externalizing Bias score, a numeric vector

`PersBias` Personal Bias score, a numeric vector

Details

The data here are for a subset of the observations in `NeuroCog` for which measures on various scales of social cognition were also available. Interest here is on whether the schizophrenia group can be distinguished from the schizoaffective group on these measures.

The Social Cognitive measures were designed to tap various aspects of the perception and cognitive procession of emotions of others. Emotion perception was assessed using a Managing Emotions (`MgeEmotions`) score from the MCCB. A "theory of mind" (`ToM`) score assessed ability to read the emotions of others from photographs of the eye region of male and female faces. Two other measures, externalizing bias (`ExtBias`) and personalizing bias (`PersBias`) were calculated from a scale measuring the degree to which individuals attribute internal, personal or situational causal attributions to positive and negative social events.

See `NeuroCog` for a description of the sample. Only those with complete data on all the social cognitive measures are included in this data set.

There is one extreme outlier in the schizophrenia group and other possible outliers in the control group, left in here for tutorial purposes.

Source

Hartman, L. I. (2016). Schizophrenia and Schizoaffective Disorder: One Condition or Two? Unpublished PhD dissertation, York University.

Heinrichs, R.W., Pinnock, F., Muharib, E., Hartman, L.I., Goldberg, J.O., & McDermid Vaz, S. (2015). Neurocognitive normality in schizophrenia revisited. *Schizophrenia Research: Cognition*, **2** (4), 227-232. doi: 10.1016/j.scog.2015.09.001

Examples

```
library(car)
data(SocialCog)
SC.mod <- lm(cbind(MgeEmotions, ToM, ExtBias, PersBias) ~ Dx, data=SocialCog)
SC.mod
```

```
car::Anova(SC.mod)

# test hypotheses of interest in terms of contrasts
print(linearHypothesis(SC.mod, "Dx1"), SSP=FALSE)
print(linearHypothesis(SC.mod, "Dx2"), SSP=FALSE)

#' ## HE plots

heplot(SC.mod, hypotheses=list("Dx1"="Dx1", "Dx2"="Dx2"),
 fill=TRUE, fill.alpha=.1)

pairs(SC.mod, fill=c(TRUE,FALSE), fill.alpha=.1)
```

`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, ...)
```

Arguments

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

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

base::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])
```

---

stdcoef *Standardized Regression Coefficients*

Description

Computes standardized ("beta") regression coefficients for a fitted `lm` or `mlm` object, i.e. the coefficients that would result from fitting the same model with all numeric variables rescaled to mean 0, SD 1.

For an `"mlm"` object, standardization follows `stdmodel()`'s convention: the response(s) and numeric predictors are standardized; factor predictors are left raw.

Usage

```
stdcoef(object, ...)

## S3 method for class 'lm'
stdcoef(object, ...)

## S3 method for class 'mlm'
stdcoef(object, ...)
```

Arguments

object	A fitted "lm" or "mlm" object
...	Additional arguments. Not used.

Value

For an "lm" object, a named numeric vector of standardized coefficients (intercept excluded). For an "mlm" object, a numeric matrix of standardized coefficients (parameters x responses, intercept row excluded).

Author(s)

Michael Friendly

See Also

stdmodel(), coefplot.mlm(). For standard errors, test statistics and p-values on the standardized scale, run lmtest::coeftest() (and, for a tidy data frame, broom::tidy.coeftest()) on stdmodel()'s result.

Other multivariate linear models: coefplot(), glance.mlm(), stdmodel()

Examples

```
data(Prestige, package = "carData")
prestige.mod <- lm(prestige ~ income + education, data = Prestige)
stdcoef(prestige.mod)

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

---

stdmodel *Standardize a Fitted Linear Model*

Description

stdmodel() refits a fitted lm/mlm object on standardized (mean 0, SD 1) versions of its response variable(s) and its numeric predictors, leaving factor-coded predictors on their original (raw 0/1 dummy) scale. This is the engine behind stdcoef() for multivariate models, and behind coefplot.mlm(..., std = TRUE).

Since the result is just an ordinary refit "lm"/"mlm" object, it also works with other tools that expect one, e.g. lmtest::coeftest() and broom::tidy.coeftest() on its result.

Usage

```
stdmodel(object, ...)

## S3 method for class 'lm'
stdmodel(object, ...)
```

Arguments

`object`	A fitted `"lm"` or `"mlm"` object
`...`	Additional arguments. Not used.

Details

Only the response(s) and *numeric* predictors are standardized; factor predictors are left as-is, since "one SD" isn't a meaningful unit for a 0/1 dummy variable. This still gives an interpretable, commonly-used effect size for factor predictors: "SD units of y per unit change in the predictor."

Interactions and main effects of raw variables (e.g. `SES * (n + s)`) are handled correctly, since `stats::model.frame()` reduces a formula down to its base variables before any interaction terms are built – refitting on the standardized base variables reconstructs the interaction terms correctly.

Computed/transformed predictor terms (e.g. `log(x)`, `poly(x, 2)`) are **not** supported and raise an error, since standardizing the *already computed* column would not do what a user expects once the model is refit and the original expression is re-evaluated. Similarly, a computed univariate response (e.g. `log(y) ~ ...`) is not supported. Pre-compute these if you want to use such variables.
`subset=`/`weights=`/`na.action=` in the original fitting call are not specially handled – `stdmodel()` is designed for models fit directly via `lm(formula, data = ...)`.

Value

The refit model, of the same class as `object`

Author(s)

Michael Friendly

See Also

`stdcoef()`, `coefplot.mlm()`, `lmtest::coeftest()`, `broom::tidy.coeftest()`

Other multivariate linear models: `coefplot()`, `glance.mlm()`, `stdcoef()`

Examples

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

rohwer.std <- stdmodel(rohwer.mod)
coef(rohwer.std)
```

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

mod	An mlm model object
term	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.
label.factors	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 ':'
abbrev.levels	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. Rows are in the order of the factor levels, with the first factor varying fastest. Cells with no observations are omitted.

Author(s)

Michael Friendly

See Also

stats::aggregate(), base::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")
```

`text_usr` *Add Text to a Plot at Normalized Device Coordinates*

Description

`text_usr()` draws the strings given in the vector labels at the coordinates given by `x` and `y`, but using normalized device coordinates (0, 1) to position text at absolute locations in a plot. This is useful when you know where in a plot you want to add some text annotation, but don't want to figure out what the data coordinates are.

Usage

```
text_usr(x, y, labels, ...)
```

Arguments

`x`, `y`	numeric vectors of coordinates in (0, 1) where the text `labels` should be written. If the length of `x` and `y` differs, the shorter one is recycled. Alternatively, a single argument `x` can be provided.
`labels`	a character vector or `base::expression()` specifying the text to be written
`...`	other arguments passed to `graphics::text()`, such as `pos`, `cex`, `col`, ...

Details

`y` may be missing since `grDevices::xy.coords()` is used for construction of the coordinates.

The function also works with `par(xlog) == TRUE` and `par(ylog) == TRUE` when either of these is set for log scales.

Source

From https://stackoverflow.com/questions/25450719/plotting-text-in-r-at-absolute-position

Examples

```
library(heplots)
x = c(0.5, rep(c(0.05, 0.95), 2))
y = c(0.5, rep(c(0.05, 0.95), each=2))
plot(x, y, pch = 16,
     xlim = c(0,1),
     ylim = c(0,1))
text_usr(0.05, 0.95, "topleft",    pos = 4)
text_usr(0.95, 0.95, "topright",   pos = 2)
text_usr(0.05, 0.05, "bottomleft", pos = 4)
text_usr(0.95, 0.05, "bottomright",pos = 2)
```

TIPI *Data on the Ten Item Personality Inventory*

Description

The Ten Item Personality Inventory (Gosling et al. 2003) is a brief inventory of the Big Five personality domains (Extraversion, Neuroticism, Conscientiousness, Agreeableness, and Openness to experience). This dataset, originally from the Open Source Psychometrics Project (https://openpsychometrics.org/), was used by Jones et al. (2020), from which we obtained this version.

Format

A data frame with 1799 observations on the following 16 variables.

`Extraversion` a numeric vector

`Neuroticism` a numeric vector

`Conscientiousness` a numeric vector

`Agreeableness` a numeric vector

`Openness` a numeric vector

`education` an ordered factor with levels `<HS` < `HS` < `Univ` < `Grad`

`urban` an ordered factor with levels `Rural` < `Suburban` < `Urban`

`gender` a factor with levels `M` `F`

`engnat` a factor with levels `Native` `Non-native`

`age` a numeric vector

`religion` a factor with levels `Agnostic` `Atheist` `Buddhist` `Christian (Catholic)` `Christian (Mormon)` `Christian (Protestant)` `Christian (Other)` `Hindu` `Jewish` `Muslim` `Sikh` `Other`

`orientation` a factor with levels `Heterosexual` `Bisexual` `Homosexual` `Asexual` `Other`

`race` a factor with levels `Asian` `Arab` `Black` `Indig-White` `Other`

`voted` a factor with levels `Yes` `No`

`married` a factor with levels `Never married` `Currently married` `Previously married`

`familysize` a numeric vector

Details

In addition to scores on the Big Five scales, the dataset contains 11 demographic variables on the participants, potentially useful in multivariate analyses.

Scores on each personality domain were calculated by averaging items assigned to each domain (after reverse scoring specific items). In this version, total scores for each scale were calculated by averaging the positively and negatively coded items, for example, `TIPI$Extraversion <- (TIPI$E + (8-TIPI$E_r))/2`.

Then, for the present purposes, some tidying was done:

- 100 cases with `gender=="Other" were deleted; \item codes for levels of education, engnatandrace‘` were abbreviated for ease of use in graphics.

Source

Jones, P.J., Mair, P., Simon, T. et al. (2020). Network Trees: A Method for Recursively Partitioning Covariance Structures. *Psychometrika*, **85**, 926?945. doi:10.1007/s11336020097314.

References

Gosling, S. D., Rentfrow, P. J., & Swann, W. B, Jr. (2003). A very brief measure of the Big-Five personality domains. *Journal of Research in Personality*, **37**, 504?528.

Examples

```
data(TIPI)
# fit an mlm
tipi.mlm <- lm(cbind(Extraversion, Neuroticism, Conscientiousness, Agreeableness, Openness)
               ~ engnat + gender + education, data = TIPI )
car::Anova(tipi.mlm)

heplot(tipi.mlm, fill=TRUE, fill.alpha=0.1)

pairs(tipi.mlm, fill=TRUE, fill.alpha=0.1)

# candisc works best for factors with >2 levels
library(candisc)
tipi.can <- candisc(tipi.mlm, term="education")
tipi.can

heplot(tipi.can, fill=TRUE, fill.alpha=0.1,
       var.col = "darkred", var.cex = 1.5, var.lwd = 3)
```

`traceCI` *Calculate Confidence Interval for Trace of Covariance Matrices*

Description

This function uses asymptotic results from random matrix theory to calculate approximate, normal theory confidence intervals for the trace (sum of eigenvalues) of one or more sample covariance matrices.

The trace of a covariance matrix equals the sum of its eigenvalues, which represents the total variance in the system. For large samples, the trace follows an approximate normal distribution with variance that can be calculated from the eigenvalue structure.

Usage

```
traceCI(cov, n, conf = 0.95)
```

Arguments

`cov`	A covariance matrix or a (named) list of covariance matrices, all the same size
`n`	Sample size, or vector of sample sizes, one for each covariance matrix
`conf`	Confidence level (default: 0.95)

Details

The confidence interval is based on the asymptotic normality of the trace:

$$trace(\widehat{\Sigma}) \pm z_{1-\alpha/2} \times SE$$

where $\widehat{\Sigma}$ is the sample covariance matrix and SE is the standard error.

The variance of the trace is calculated using the formula from Bai & Silverstein (2004):

$$Var(trace) = \frac{2}{n}\sum_{i=1}^{p}\lambda_i^2 = \frac{2}{n}trace(\Sigma^2)$$

where λ_i are the eigenvalues of Σ.

This is a simplified version of the general CLT for linear spectral statistics. For i.i.d. components with finite fourth moments, the trace is asymptotically normal with this variance.

Asymptotic regime: The theory applies when both sample size n and dimension p can grow, typically requiring $n >> p$ for good finite-sample performance. The approximation improves as the sample size increases.

Comparison with bootstrap: For small to moderate samples, bootstrap confidence intervals (see `eigstatCI()`) may provide better coverage. This asymptotic approach is faster but may be anti-conservative (too narrow) when n is small relative to p.

Value

A data frame with one row for each covariance matrix. Columns:

trace The trace (sum of eigenvalues) of the covariance matrix

se Standard error of the trace

lower Lower confidence limit

upper Upper confidence limit

Author(s)

Michael Friendly

References

Bai, Z. D., & Silverstein, J. W. (2004). CLT for linear spectral statistics of large-dimensional sample covariance matrices. *Annals of Probability*, 32(1A), 553-605. doi:10.1214/aop/1078415845

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis* (3rd ed.). Wiley-Interscience.

See Also

`boxM()`, `plot.boxM()`, `logdetCI()`, `eigstatCI()`

Examples

```
# Iris data example
data(iris)
iris.mod <- lm(as.matrix(iris[,1:4]) ~ iris$Species)
iris.boxm <- boxM(iris.mod)

# Get covariance matrices
cov <- c(iris.boxm$cov, list(pooled = iris.boxm$pooled))
n <- c(rep(50, 3), 150)

# Calculate trace CIs
CI <- traceCI(cov, n = n, conf = 0.95)
CI

# Compare with plot
plot(iris.boxm, which = "sum", gplabel = "Species",
     main = "Sum of eigenvalues (trace)")
arrows(CI$lower, 1:4, CI$upper, 1:4,
       lwd = 3, angle = 90, length = 0.1, code = 3, col = "red")

# Single covariance matrix
S <- cov(iris[,1:4])
traceCI(S, n = 150)

# Compare different confidence levels
traceCI(cov, n = n, conf = 0.90)
```

```
traceCI(cov, n = n, conf = 0.95)
traceCI(cov, n = n, conf = 0.99)
```

`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

%% ~~objects to See Also as `utils::help()`, ~~~ `grDevices::col2rgb()`, `grDevices::rgb()`, `grDevices::adjustcolor()`,

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=""))
```

---

uniStats *Univariate Test Statistics for a Multivariate Linear Model*

Description

Univariate Test Statistics for a Multivariate Linear Model

Usage

```
uniStats(x, ...)
```

Arguments

x	A "mlm" object fitted by stats::lm() with two or more response variables
...	Other arguments, ignored

Value

An object of class c("anova", "data.frame") containing, for each response variable the overall R^2 for all terms in the model and the overall F statistic together with its degrees of freedom and p-value.

See Also

glance.mlm()

Examples

```
iris.mod <- lm(cbind(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) ~ Species, data=iris)
car::Anova(iris.mod)
uniStats(iris.mod)

data(Plastic, package = "heplots")
plastic.mod <- lm(cbind(tear, gloss, opacity) ~ rate*additive, data=Plastic)
# multivariate tests
car::Anova(plastic.mod)
```

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.

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. doi:10.18637/jss.v037.i04.

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

Examples

```
library(car)
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))
car::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, iterm="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.

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. doi:10.18637/jss.v037.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:
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 <- car::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]) %*% measure

between.mod <- lm(between ~ group, data=WeightLoss)
car::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]) %*% month
within.mod <- lm(trends ~ group, data=WeightLoss)
car::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

∗ **3D plotting**
arrow3d, 9
bbox3d, 11
cross3d, 30
ellipse3d.axes, 44
heplot3d, 65
∗ **HE plot functions**
heplot, 54
heplot1d, 61
heplot3d, 65
pairs.mlm, 101
∗ **HE plots**
ReadingDisability, 122
∗ **MANCOVA**
LearnDis, 81
Rohwer, 129
∗ **MANOVA**
AddHealth, 6
Bees, 12
Diabetes, 31
dogfood, 36
Headache, 53
Iwasaki_Big_Five, 77
LearnDis, 81
mathscore, 88
MockJury, 89
NeuroCog, 91
oral, 96
Oslo, 98
Parenting, 103
Plastic, 106
Pottery2, 113
Probe, 115
RatWeight, 119
ReadingDisability, 122
RootStock, 131
schoolsites, 136
Skulls, 137
SocGrades, 139
SocialCog, 140
TIPI, 148
∗ **MMRA**
Adopted, 7
FootHead, 49
Hernior, 71
NLSY, 93
Overdose, 99
Rohwer, 129
Sake, 132
schooldata, 134
∗ **aplot**
arrow3d, 9
cross3d, 30
ellipse3d.axes, 44
heplot, 54
heplot1d, 61
heplot3d, 65
heplots-package, 4
mark.H0, 87
∗ **cancor**
Overdose, 99
∗ **candisc**
dogfood, 36
Hernior, 71
MockJury, 89
NeuroCog, 91
Oslo, 98
Pottery2, 113
SocGrades, 139
SocialCog, 140
TIPI, 148
∗ **color**
trans.colors, 152
∗ **contrasts**
dogfood, 36
FootHead, 49
Probe, 115
RootStock, 131

Skulls, 137
VocabGrowth, 154
∗ **contrast**
NeuroCog, 91
∗ **covariance ellipses**
covEllipses, 22
ellipse.axes, 40
ellipse.box, 42
ellipse3d.axes, 44
∗ **datasets**
AddHealth, 6
Adopted, 7
Bees, 12
Diabetes, 31
dogfood, 36
FootHead, 49
Headache, 53
Hernior, 71
Iwasaki_Big_Five, 77
LearnDis, 81
mathscore, 88
MockJury, 89
NeuroCog, 91
NLSY, 93
oral, 96
Oslo, 98
Overdose, 99
Parenting, 103
peng, 104
Plastic, 106
Pottery2, 113
Probe, 115
RatWeight, 119
ReactTime, 121
ReadingDisability, 122
Rohwer, 129
RootStock, 131
Sake, 132
schooldata, 134
schoolsites, 136
Skulls, 137
SocGrades, 139
SocialCog, 140
TIPI, 148
VocabGrowth, 154
WeightLoss, 155
∗ **diagnostic plots**
cqplot, 26
distancePlot, 33
eigstatCI, 38
plot.boxM, 107
∗ **dynamic**
cross3d, 30
ellipse3d.axes, 44
heplot3d, 65
∗ **effect size**
etasq, 47
∗ **hgraph**
plot.boxM, 107
∗ **homogeneity tests**
bartlettTests, 10
leveneTests, 82
∗ **hplot**
coefplot, 17
covEllipses, 22
cqplot, 26
heplot, 54
heplot1d, 61
heplot3d, 65
heplots-package, 4
interpPlot, 73
pairs.mlm, 101
plot.robmlm, 109
∗ **htest**
bartlettTests, 10
leveneTests, 82
∗ **manip**
colDevs, 20
gsorth, 51
logdetCI, 84
stdcoef, 143
stdmodel, 144
termMeans, 146
traceCI, 150
∗ **models**
termMeans, 146
∗ **multivariate linear models**
coefplot, 17
glance.mlm, 50
stdcoef, 143
stdmodel, 144
∗ **multivariate**
covEllipses, 22
cqplot, 26
etasq, 47
glance.mlm, 50

heplot, 54
heplot1d, 61
heplot3d, 65
heplots-package, 4
Mahalanobis, 86
pairs.mlm, 101
robmlm, 125
statList, 142
∗ **ordered**
AddHealth, 6
∗ **package**
heplots-package, 4
∗ **repeated**
Adopted, 7
Headache, 53
Probe, 115
RatWeight, 119
ReactTime, 121
VocabGrowth, 154
∗ **robust methods**
Mahalanobis, 86
plot.robmlm, 109
robmlm, 125
∗ **robust**
NLSY, 93
robmlm, 125
schooldata, 134
∗ **stepdown analysis**
LearnDis, 81
∗ **strength of association**
etasq, 47
∗ **utilities**
noteworthy, 94
statList, 142
text_usr, 147
_PACKAGE *(*heplots-package*)*, 4

AddHealth, 6
Adopted, 7
animation::animation, *74*
archdata::RBPottery, *114*
arrow3d, 9
arrow3d(), *12*, *31*, *45*, *70*

bartlettTests, 10
bartlettTests(), *83*
base::chol(), *43*
base::colMeans(), *21*, *143*, *146*
base::expression(), *147*
base::interaction(), *20*
base::options(), *126*
base::qr(), *52*
base::sweep(), *21*
bbox3d, 11
bbox3d(), *10*, *31*, *45*, *70*
Bees, 12
boxM, 14
boxM(), *11*, *25*, *39*, *85*, *108*
broom::glance.lm(), *50*
broom::tidy.coeftest(), *144, 145*

candisc::candisc(), *59, 60*
candisc::heplot.candisc(), *59, 60*
candisc::heplot3d.candisc(), *59*, *70*
car::Anova(), *4*, *47, 48*, *57*, *60*, *63*, *65*, *67*, *70*, *102*, *125*
car::avPlots(), *118*
car::confidenceEllipse(), *19, 20*
car::dataEllipse(), *74*, *118*
car::ellipse(), *79, 80*
car::influencePlot(), *33*
car::leveneTest(), *16*, *82, 83*
car::linearHypothesis(), *4*, *56*, *59, 60*, *62*, *65*, *67*, *70*
car::qqPlot(), *29*
car::showLabels(), *28*, *74*, *94*, *118*
carData::Pottery, *113, 114*
coefplot, 17
coefplot(), *51*, *144, 145*
coefplot.mlm(), *60*, *78*, *144, 145*
colDevs, 20
colMeansList *(*statList*)*, 142
colMeansList(), *146*
covEllipses, 22
covEllipses(), *15, 16*, *42, 43*, *45*, *78*, *80*
covList *(*statList*)*, 142
cqplot, 26
cqplot(), *35*, *39*, *108*
cross3d, 30
cross3d(), *10*, *12*, *45*, *70*, *88*

df.terms, 31
Diabetes, 31
distancePlot, 33
distancePlot(), *29*, *39*, *108*
dogfood, 36

effectsize::eta_squared(), *48*

eigstatCI, 38
eigstatCI(), *29*, *35*, *108*
eigstatCI_boxM *(*eigstatCI*)*, 38
ellipse.axes, 40
ellipse.axes(), *25*, *43*, *45*
ellipse.box, 42
ellipse.box(), *25*, *42*, *45*
ellipse3d.axes, 44
ellipse3d.axes(), *10*, *12*, *25*, *31*, *42, 43*, *70*
Ellipsoid, 45
etasq, 47

FootHead, 49

glance.mlm, 50
glance.mlm(), *20*, *144, 145*, *153*
graphics::dotchart(), *108*
graphics::lines(), *41–43*
graphics::pairs(), *101*
graphics::par(), *27*
graphics::plot(), *18*, *110*
graphics::text(), *34*, *41, 42*, *79, 80*, *147*
grDevices::adjustcolor(), *152*
grDevices::col2rgb(), *152*
grDevices::palette(), *27*
grDevices::rgb(), *152*
grDevices::xy.coords(), *147*
gsorth, 51

Headache, 53
heplot, 54
heplot(), *25*, *64, 65*, *68–70*, *78*, *80*, *101–103*, *125*
heplot1d, 61
heplot1d(), *60*, *70*, *103*
heplot3d, 65
heplot3d(), *10*, *12*, *31*, *45*, *59, 60*, *65*, *103*
heplots *(*heplots-package*)*, 4
heplots-package, 4
Hernior, 71

influencePlot(), *33*
interpPlot, 73
Iwasaki_Big_Five, 77

knitr::include_graphics(), *70*

label.ellipse, 78
label.ellipse(), *19*, *24*, *56*, *60*
LearnDis, 81

leveneTests, 82
leveneTests(), *11*
lmtest::coeftest(), *144, 145*
logdetCI, 84
logdetCI(), *39*, *108*

Mahalanobis, 86
Mahalanobis(), *29*, *35*, *110*, *127*
mark.H0, 87
mark.H0(), *57*, *60*
MASS::cov.rob(), *22*, *25*, *86, 87*
MASS::cov.trob(), *125*, *127*
MASS::psi.bisquare(), *126*
MASS::rlm(), *125*, *127*
mathscore, 88
MockJury, 89

NeuroCog, 91, *141*
NLSY, 93
noteworthy, 94

oral, 96
Oslo, 98
Overdose, 99

pairs.mlm, 101
pairs.mlm(), *59, 60*, *65*, *70*
parameters::model_parameters(), *17*, *20*
Parenting, 103
peng, 104
Plastic, 106
plot.boxM, 107
plot.boxM(), *16*, *29*, *35*, *39*, *85*
plot.robmlm, 109
plot.robmlm(), *87*, *125*, *127*
plot_boxM_boot, 111
Pottery2, 113
print.boxM *(*boxM*)*, 14
print.robmlm *(*robmlm*)*, 125
print.summary.robmlm *(*robmlm*)*, 125
Probe, 115
Probe1 *(*Probe*)*, 115
Probe2 *(*Probe*)*, 115
pvPlot, 117

qqtest::qqtest(), *29*

RatWeight, 119
ReactTime, 121
ReadingDisability, 122

rel_diff, 123
rgl::bg3d(), *68*
rgl::ellipse3d(), *45*
rgl::lines3d(), *10*
rgl::material3d(), *9*
rgl::movie3d(), *69*
rgl::play3d(), *69*
rgl::points3d(), *88*
rgl::rglwidget(), *69*
rgl::segments3d(), *10*, *30, 31*, *45*
rgl::setupKnitr(), *69*
rgl::shade3d(), *68*
rgl::text3d(), *45*
rgl::view3d(), *68*
rgl::wire3d(), *68*
robmlm, 125
robmlm(), *33*, *35*, *87*, *109, 110*, *124*
Rohwer, 129
RootStock, 131
rrcov::OsloTransect, *98*

Sake, 132
schooldata, 134, *136, 137*
schoolsites, *134, 135*, 136
Skulls, 137
Sleuth2::ex1605, *8*
SocGrades, 139
SocialCog, 140
statList, 142
statList(), *146*
stats::aggregate(), *146*
stats::anova(), *52*
stats::bartlett.test(), *10*
stats::coefficients(), *127*
stats::effects(), *127*
stats::fitted.values(), *127*
stats::lm(), *26*, *31*, *34*, *50*, *127*, *153*
stats::mahalanobis(), *86, 87*
stats::model.frame(), *145*
stats::na.omit(), *126*
stats::qqplot(), *29*
stats::residuals(), *127*
stats::update(), *126*
stats::vcov(), *125*
stdcoef, 143
stdcoef(), *20*, *51*, *144, 145*
stdmodel, 144
stdmodel(), *19, 20*, *51*, *143, 144*
summary.boxM *(*boxM*)*, 14
summary.robmlm *(*robmlm*)*, 125

termMeans, 146
termMeans(), *143*
text_usr, 147
tibble::tibble(), *50, 51*
TIPI, 148
traceCI, 150
trans.colors, 152
trans.colors(), *60*

uniStats, 153
utils::help(), *152*

VocabGrowth, 154

WeightLoss, 155