Package ‘multcomp’

September 23, 2020

Title Simultaneous Inference in General Parametric Models

Version 1.4-14

Date 2020-09-23

Description Simultaneous tests and confidence intervals
for general linear hypotheses in parametric models, including
linear, generalized linear, linear mixed effects, and survival models.
The package includes demos reproducing analyzes presented
in the book ``Multiple Comparisons Using R'' (Bretz, Hothorn,
Westfall, 2010, CRC Press).

Depends stats, graphics, mvtnorm (>= 1.0-10), survival (>= 2.39-4),
TH.data (>= 1.0-2)

Imports sandwich (>= 2.3-0), codetools

Suggests lme4 (>= 0.999375-16), nlme, robustbase, coin, MASS, foreign,
xtable, lmtest, coxme (>= 2.2-1), SimComp, ISwR, tram (>=
0.2-5)

URL http://multcomp.R-forge.R-project.org The publishers web page is

http://www.crcpress.com/product/isbn/9781584885740

LazyData yes

License GPL-2

NeedsCompilation no

Author Torsten Hothorn [aut, cre] (<https://orcid.org/0000-0001-8301-0471>),
Frank Bretz [aut],
Peter Westfall [aut],
Richard M. Heiberger [ctb],
Andre Schuetzenmeister [ctb],
Susan Scheibe [ctb]

Maintainer Torsten Hothorn <Torsten.Hothorn@R-project.org>

Repository CRAN

Date/Publication 2020-09-23 15:10:03 UTC

R topics documented:

`adevent` *Adverse Events Data*

Description

Indicators of 28 adverse events in a two-arm clinical trial.

Usage

```
data(adevent)
```

Format

A data frame with 160 observations on the following 29 variables.

`E1` a factor with levels `no event` `event`

`E2` a factor with levels `no event` `event`

`E3` a factor with levels `no event` `event`

`E4` a factor with levels `no event` `event`

`E5` a factor with levels `no event` `event`

`E6` a factor with levels `no event` `event`

`E7` a factor with levels `no event` `event`

`E8` a factor with levels `no event` `event`

`E9` a factor with levels `no event` `event`

`E10` a factor with levels `no event` `event`

`E11` a factor with levels `no event` `event`

`E12` a factor with levels `no event` `event`

`E13` a factor with levels `no event` `event`

`E14` a factor with levels `no event` `event`

`E15` a factor with levels `no event` `event`

`E16` a factor with levels `no event` `event`

`E17` a factor with levels `no event` `event`

`E18` a factor with levels `no event` `event`

`E19` a factor with levels `no event` `event`

`E20` a factor with levels `no event` `event`

`E21` a factor with levels `no event` `event`

`E22` a factor with levels `no event` `event`

`E23` a factor with levels `no event` `event`

`E24` a factor with levels `no event` `event`

`E25` a factor with levels `no event` `event`

`E26` a factor with levels `no event` `event`

`E27` a factor with levels `no event` `event`

`E28` a factor with levels `no event` `event`

`group` group indicator.

Details

The data is provided by Westfall et al. (1999, p. 242) and contains binary indicators of 28 adverse events (`E1`,..., `E28`) for two arms (`group`).

Source

P. H. Westfall, R. D. Tobias, D. Rom, R. D. Wolfinger, Y. Hochberg (1999). *Multiple Comparisons and Multiple Tests Using the SAS System*. Cary, NC: SAS Institute Inc.

cftest *Testing Estimated Coefficients*

Description

A convenience function for univariate testing via z- and t-tests of estimated model coefficients

Usage

```
cftest(model, parm, test = univariate(), ...)
```

Arguments

model	a fitted model.
parm	a vector of parameters to be tested, either a character vector of names or an integer.
test	a function for computing p values, see summary.glht.
...	additional arguments passed to summary.glht.

Details

The usual z- or t-tests are tested without adjusting for multiplicity.

Value

An object of class summary.glht.

See Also

coeftest

Examples

```
lmod <- lm(dist ~ speed, data = cars)
summary(lmod)
cftest(lmod)
```

`cholesterol` *Cholesterol Reduction Data Set*

Description

Cholesterol reduction for five treatments.

Usage

```
data("cholesterol")
```

Format

This data frame contains the following variables

trt treatment groups, a factor at levels `1time`, `2times`, `4times`, `drugD` and `drugE`.

response cholesterol reduction.

Details

A clinical study was conducted to assess the effect of three formulations of the same drug on reducing cholesterol. The formulations were 20mg at once (`1time`), 10mg twice a day (`2times`), and 5mg four times a day (`4times`). In addition, two competing drugs were used as control group (`drugD` and `drugE`). The purpose of the study was to find which of the formulations, if any, is efficacious and how these formulations compare with the existing drugs.

Source

P. H. Westfall, R. D. Tobias, D. Rom, R. D. Wolfinger, Y. Hochberg (1999). *Multiple Comparisons and Multiple Tests Using the SAS System*. Cary, NC: SAS Institute Inc., page 153.

Examples

```
### adjusted p-values for all-pairwise comparisons in a one-way layout
### set up ANOVA model
amod <- aov(response ~ trt, data = cholesterol)

### set up multiple comparisons object for all-pair comparisons
cht <- glht(amod, linfct = mcp(trt = "Tukey"))

### cf. Westfall et al. (1999, page 171)
summary(cht, test = univariate())
summary(cht, test = adjusted("Shaffer"))
summary(cht, test = adjusted("Westfall"))

### use only a subset of all pairwise hypotheses
K <- contrMat(table(cholesterol$trt), type="Tukey")
Ksub <- rbind(K[c(1,2,5),],
```

```
               "D - test" = c(-1, -1, -1, 3, 0),
               "E - test" = c(-1, -1, -1, 0, 3))

### reproduce results in Westfall et al. (1999, page 172)
### note: the ordering of our estimates here is different
amod <- aov(response ~ trt - 1, data = cholesterol)
summary(glht(amod, linfct = mcp(trt = Ksub[,5:1])),
        test = adjusted("Westfall"))
```

`cld` *Set up a compact letter display of all pair-wise comparisons*

Description

Extract information from `glht`, `summary.glht` or `confint.glht` objects which is required to create and plot compact letter displays of all pair-wise comparisons.

Usage

```
## S3 method for class 'summary.glht'
cld(object, level = 0.05, decreasing = FALSE, ...)
## S3 method for class 'glht'
cld(object, level = 0.05, decreasing = FALSE, ...)
## S3 method for class 'confint.glht'
cld(object, decreasing = FALSE, ...)
```

Arguments

`object`	An object of class `glht`, `summary.glht` or `confint.glht`.
`level`	Significance-level to be used to term a specific pair-wise comparison significant.
`decreasing`	logical. Should the order of the letters be increasing or decreasing?
`...`	additional arguments.

Details

This function extracts all the information from `glht`, `summary.glht` or `confint.glht` objects that is required to create a compact letter display of all pair-wise comparisons. In case the contrast matrix is not of type `"Tukey"`, an error is issued. In case of `confint.glht` objects, a pair-wise comparison is termed significant whenever a particular confidence interval contains 0. Otherwise, p-values are compared to the value of `"level"`. Once, this information is extracted, plotting of all pair-wise comparisons can be carried out.

Value

An object of class `cld`, a list with items:

y	Values of the response variable of the original model.
yname	Name of the response variable.
x	Values of the variable used to compute Tukey contrasts.
weights	Weights used in the fitting process.
lp	Predictions from the fitted model.
covar	A logical indicating whether the fitted model contained covariates.
signif	Vector of logicals indicating significant differences with hyphenated names that identify pair-wise comparisons.

References

Hans-Peter Piepho (2004), An Algorithm for a Letter-Based Representation of All-Pairwise Comparisons, *Journal of Computational and Graphical Statistics*, **13**(2), 456–466.

See Also

glht plot.cld

Examples

```
### multiple comparison procedures
### set up a one-way ANOVA
data(warpbreaks)
amod <- aov(breaks ~ tension, data = warpbreaks)
### specify all pair-wise comparisons among levels of variable "tension"
tuk <- glht(amod, linfct = mcp(tension = "Tukey"))
### extract information
tuk.cld <- cld(tuk)
### use sufficiently large upper margin
old.par <- par(mai=c(1,1,1.25,1), no.readonly = TRUE)
### plot
plot(tuk.cld)
par(old.par)

### now using covariates
data(warpbreaks)
amod2 <- aov(breaks ~ tension + wool, data = warpbreaks)
### specify all pair-wise comparisons among levels of variable "tension"
tuk2 <- glht(amod2, linfct = mcp(tension = "Tukey"))
### extract information
tuk.cld2 <- cld(tuk2)
### use sufficiently large upper margin
old.par <- par(mai=c(1,1,1.25,1), no.readonly = TRUE)
### plot using different colors
plot(tuk.cld2, col=c("black", "red", "blue"))
par(old.par)
```

```
### set up all pair-wise comparisons for count data
data(Titanic)
mod <- glm(Survived ~ Class, data = as.data.frame(Titanic), weights = Freq, family = binomial())
### specify all pair-wise comparisons among levels of variable "Class"
glht.mod <- glht(mod, mcp(Class = "Tukey"))
### extract information
mod.cld <- cld(glht.mod)
### use sufficiently large upper margin
old.par <- par(mai=c(1,1,1.5,1), no.readonly = TRUE)
### plot
plot(mod.cld)
par(old.par)
```

`cml` *Chronic Myelogenous Leukemia survival data.*

Description

Survival in a randomised trial comparing three treatments for Chronic Myelogeneous Leukemia (simulated data).

Usage

```
data("cml")
```

Format

A data frame with 507 observations on the following 7 variables.

`center` a factor with 54 levels indicating the study center.

`treatment` a factor with levels `trt1`, `trt2`, `trt3` indicating the treatment group.

`sex` sex (0 = female, 1 = male)

`age` age in years

`riskgroup` risk group (0 = low, 1 = medium, 2 = high)

`status` censoring status (FALSE = censored, TRUE = dead)

`time` survival or censoring time in days.

Details

The data are simulated according to structure of the data by the German CML Study Group used in Hehlmann (1994).

Source

R. Hehlmann, H. Heimpel, J. Hasford, H.J. Kolb, H. Pralle, D.K. Hossfeld, W. Queisser, H. Loeffler, A. Hochhaus, B. Heinze (1994), Randomized comparison of interferon-alpha with busulfan and hydroxyurea in chronic myelogenous leukemia. The German CML study group. *Blood* **84**(12):4064-4077.

Examples

```
if (require("coxme")) {
    data("cml")
    ### one-sided simultaneous confidence intervals for many-to-one
    ### comparisons of treatment effects concerning time of survival
    ### modeled by a frailty Cox model with adjustment for further
    ### covariates and center-specific random effect.
   cml_coxme <- coxme(Surv(time, status) ~ treatment + sex + age + riskgroup + (1|center),
                       data = cml)
    glht_coxme <- glht(model = cml_coxme, linfct = mcp(treatment = "Dunnett"),
                       alternative = "greater")
    ci_coxme <- confint(glht_coxme)
    exp(ci_coxme$confint)[1:2,]
}
```

---

contrMat *Contrast Matrices*

Description

Computes contrast matrices for several multiple comparison procedures.

Usage

```
contrMat(n, type = c("Dunnett", "Tukey", "Sequen", "AVE",
                     "Changepoint", "Williams", "Marcus",
                     "McDermott", "UmbrellaWilliams", "GrandMean"),
         base = 1)
```

Arguments

n	a (possibly named) vector of sample sizes for each group.
type	type of contrast.
base	an integer specifying which group is considered the baseline group for Dunnett contrasts.

Details

Computes the requested matrix of contrasts for comparisons of mean levels.

Value

The matrix of contrasts with appropriate row names is returned.

References

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

Frank Bretz, Alan Genz and Ludwig A. Hothorn (2001), On the numerical availability of multiple comparison procedures. *Biometrical Journal*, **43**(5), 645–656.

Examples

```
n <- c(10,20,30,40)
names(n) <- paste("group", 1:4, sep="")
contrMat(n) # Dunnett is default
contrMat(n, base = 2) # use second level as baseline
contrMat(n, type = "Tukey")
contrMat(n, type = "Sequen")
contrMat(n, type = "AVE")
contrMat(n, type = "Changepoint")
contrMat(n, type = "Williams")
contrMat(n, type = "Marcus")
contrMat(n, type = "McDermott")
### Umbrella-protected Williams contrasts, i.e. a sequence of
### Williams-type contrasts with groups of higher order
### stepwise omitted
contrMat(n, type = "UmbrellaWilliams")
### comparison of each group with grand mean of all groups
contrMat(n, type = "GrandMean")
```

---

detergent *Detergent Durability Data Set*

Description

Detergent durability in an incomplete two-way design.

Usage

```
data("detergent")
```

Format

This data frame contains the following variables

detergent detergent, a factor at levels `A`, `B`, `C`, `D`, and `E`.

block block, a factor at levels `B_1`, ..., `B_10`.

plates response variable: number of plates washed before the foam disappears.

Details

Plates were washed with five detergent varieties, in ten blocks. A complete design would have 50 combinations, here only three detergent varieties in each block were applied in a balanced incomplete block design. Note that there are six observations taken at each detergent level.

Source

H. Scheffe (1959). *The Analysis of Variance*. New York: John Wiley & Sons, page 189.

P. H. Westfall, R. D. Tobias, D. Rom, R. D. Wolfinger, Y. Hochberg (1999). *Multiple Comparisons and Multiple Tests Using the SAS System*. Cary, NC: SAS Institute Inc., page 189.

Examples

```
  ### set up two-way ANOVA without interactions
  amod <- aov(plates ~ block + detergent, data = detergent)

  ### set up all-pair comparisons
  dht <- glht(amod, linfct = mcp(detergent = "Tukey"))

  ### see Westfall et al. (1999, p. 190)
  confint(dht)

  ### see Westfall et al. (1999, p. 192)
  summary(dht, test = univariate())
  ## Not run:
  summary(dht, test = adjusted("Shaffer"))
  summary(dht, test = adjusted("Westfall"))

## End(Not run)
```

`fattyacid` *Fatty Acid Content of Bacillus simplex.*

Description

Fatty acid content of different putative ecotypes of Bacillus simplex.

Usage

```
data("fattyacid")
```

Format

A data frame with 93 observations on the following 2 variables.

`PE` a factor with levels `PE3`, `PE4`, `PE5`, `PE6`, `PE7`, `PE9` indicating the putative ecotype (PE).

`FA` a numeric vector indicating the content of fatty acid (FA).

Details

The data give the fatty acid content for different putative ecotypes of Bacillus simplex. Variances of the values of fatty acid are heterogeneous among the putative ecotypes.

Source

J. Sikorski, E. Brambilla, R. M. Kroppenstedt, B. J. Tindal (2008), The temperature adaptive fatty acid content in Bacillus simplex strains from "Evolution Canyon", Israel. *Microbiology* **154**, 2416-2426.

Examples

```
if (require("sandwich")) {
    data("fattyacid")
    ### all-pairwise comparisons of the means of fatty acid content
    ### FA between different putative ecotypes PE accounting for
    ### heteroscedasticity by using a heteroscedastic consistent
    ### covariance estimation
    amod <- aov(FA ~ PE, data = fattyacid)
    amod_glht <- glht(amod, mcp(PE = "Tukey"), vcov = vcovHC)
    summary(amod_glht)

    ### simultaneous confidence intervals for the differences of
    ### means of fatty acid content between the putative ecotypes
    confint(amod_glht)
}
```

---

glht *General Linear Hypotheses*

Description

General linear hypotheses and multiple comparisons for parametric models, including generalized linear models, linear mixed effects models, and survival models.

Usage

```
## S3 method for class 'matrix'
glht(model, linfct,
    alternative = c("two.sided", "less", "greater"),
    rhs = 0, ...)
## S3 method for class 'character'
glht(model, linfct, ...)
## S3 method for class 'expression'
glht(model, linfct, ...)
## S3 method for class 'mcp'
glht(model, linfct, ...)
```

```
## S3 method for class 'mlf'
glht(model, linfct, ...)
mcp(..., interaction_average = FALSE, covariate_average = FALSE)
```

Arguments

`model` a fitted model, for example an object returned by `lm`, `glm`, or `aov` etc. It is assumed that `coef` and `vcov` methods are available for `model`. For multiple comparisons of means, methods `model.matrix`, `model.frame` and `terms` are expected to be available for `model` as well.

`linfct` a specification of the linear hypotheses to be tested. Linear functions can be specified by either the matrix of coefficients or by symbolic descriptions of one or more linear hypotheses. Multiple comparisons in AN(C)OVA models are specified by objects returned from function `mcp`..

`alternative` a character string specifying the alternative hypothesis, must be one of '"two.sided"' (default), '"greater"' or '"less"'. You can specify just the initial letter.

`rhs` an optional numeric vector specifying the right hand side of the hypothesis.

`interaction_average` logical indicating if comparisons are averaging over interaction terms. Experimental!

`covariate_average` logical indicating if comparisons are averaging over additional covariates. Experimental!

`...` additional arguments to function `modelparm` in all `glht` methods. For function `mcp`, multiple comparisons are defined by matrices or symbolic descriptions specifying contrasts of factor levels where the arguments correspond to factor names.

Details

A general linear hypothesis refers to null hypotheses of the form $H_0 : K\theta = m$ for some parametric model `model` with parameter estimates `coef(model)`.

The null hypothesis is specified by a linear function $K\theta$, the direction of the alternative and the right hand side m. Here, `alternative` equal to `"two.sided"` refers to a null hypothesis $H_0 : K\theta = m$, whereas `"less"` corresponds to $H_0 : K\theta \geq m$ and `"greater"` refers to $H_0 : K\theta \leq m$. The right hand side vector m can be defined via the `rhs` argument.

The generic method `glht` dispatches on its second argument (`linfct`). There are three ways, and thus methods, to specify linear functions to be tested:

1) The `matrix` of coefficients K can be specified directly via the `linfct` argument. In this case, the number of columns of this matrix needs to correspond to the number of parameters estimated by `model`. It is assumed that appropriate `coef` and `vcov` methods are available for `model` (`modelparm` deals with some exceptions).

2) A symbolic description, either a `character` or `expression` vector passed to `glht` via its `linfct` argument, can be used to define the null hypothesis. A symbolic description must be interpretable as a valid R expression consisting of both the left and right hand side of a linear hypothesis. Only the names of `coef(model)` must be used as variable names. The alternative is given by the direction

under the null hypothesis (= or == refer to "two.sided", <= means "greater" and >= indicates "less"). Numeric vectors of length one are valid values for the right hand side.

3) Multiple comparisons of means are defined by objects of class mcp as returned by the mcp function. For each factor, which is included in model as independent variable, a contrast matrix or a symbolic description of the contrasts can be specified as arguments to mcp. A symbolic description may be a character or expression where the factor levels are only used as variables names. In addition, the type argument to the contrast generating function contrMat may serve as a symbolic description of contrasts as well.

4) The lsm function in package lsmeans offers a symbolic interface for the definition of least-squares means for factor combinations which is very helpful when more complex contrasts are of special interest.

The mcp function must be used with care when defining parameters of interest in two-way ANOVA or ANCOVA models. Here, the definition of treatment differences (such as Tukey's all-pair comparisons or Dunnett's comparison with a control) might be problem specific. Because it is impossible to determine the parameters of interest automatically in this case, mcp in multcomp version 1.0-0 and higher generates comparisons for the main effects only, ignoring covariates and interactions (older versions automatically averaged over interaction terms). A warning is given. We refer to Hsu (1996), Chapter 7, and Searle (1971), Chapter 7.3, for further discussions and examples on this issue.

glht extracts the number of degrees of freedom for models of class lm (via modelparm) and the exact multivariate t distribution is evaluated. For all other models, results rely on the normal approximation. Alternatively, the degrees of freedom to be used for the evaluation of multivariate t distributions can be given by the additional df argument to modelparm specified via ....

glht methods return a specification of the null hypothesis $H_0 : K\theta = m$. The value of the linear function $K\theta$ can be extracted using the coef method and the corresponding covariance matrix is available from the vcov method. Various simultaneous and univariate tests and confidence intervals are available from summary.glht and confint.glht methods, respectively.

A more detailed description of the underlying methodology is available from Hothorn et al. (2008) and Bretz et al. (2010).

Value

An object of class glht, more specifically a list with elements

model	a fitted model, used in the call to glht
linfct	the matrix of linear functions
rhs	the vector of right hand side values m
coef	the values of the linear functions
vcov	the covariance matrix of the values of the linear functions
df	optionally, the degrees of freedom when the exact t distribution is used for inference
alternative	a character string specifying the alternative hypothesis
type	optionally, a character string giving the name of the specific procedure

with print, summary, confint, coef and vcov methods being available. When called with linfct being an mcp object, an additional element focus is available storing the names of the factors under test.

References

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

Shayle R. Searle (1971), *Linear Models*. John Wiley \& Sons, New York.

Jason C. Hsu (1996), *Multiple Comparisons*. Chapman & Hall, London.

Torsten Hothorn, Frank Bretz and Peter Westfall (2008), Simultaneous Inference in General Parametric Models. *Biometrical Journal*, **50**(3), 346–363; See `vignette("generalsiminf",package = "multcomp")`.

Examples

```
### multiple linear model, swiss data
lmod <- lm(Fertility ~ ., data = swiss)

### test of H_0: all regression coefficients are zero
### (ignore intercept)

### define coefficients of linear function directly
K <- diag(length(coef(lmod)))[-1,]
rownames(K) <- names(coef(lmod))[-1]
K

### set up general linear hypothesis
glht(lmod, linfct = K)

### alternatively, use a symbolic description
### instead of a matrix
glht(lmod, linfct = c("Agriculture = 0",
                      "Examination = 0",
                      "Education = 0",
                      "Catholic = 0",
                      "Infant.Mortality = 0"))


### multiple comparison procedures
### set up a one-way ANOVA
amod <- aov(breaks ~ tension, data = warpbreaks)

### set up all-pair comparisons for factor `tension'
### using a symbolic description (`type' argument
### to `contrMat()')
glht(amod, linfct = mcp(tension = "Tukey"))

### alternatively, describe differences symbolically
glht(amod, linfct = mcp(tension = c("M - L = 0",
                                    "H - L = 0",
                                    "H - M = 0")))

### alternatively, define contrast matrix directly
contr <- rbind("M - L" = c(-1, 1, 0),
```

```
                 "H - L" = c(-1, 0, 1),
                 "H - M" = c(0, -1, 1))
glht(amod, linfct = mcp(tension = contr))

### alternatively, define linear function for coef(amod)
### instead of contrasts for `tension'
### (take model contrasts and intercept into account)
glht(amod, linfct = cbind(0, contr %*% contr.treatment(3)))


### mix of one- and two-sided alternatives
warpbreaks.aov <- aov(breaks ~ wool + tension,
                    data = warpbreaks)

### contrasts for `tension'
K <- rbind("L - M" = c( 1, -1,  0),
           "M - L" = c(-1,  1,  0),
           "L - H" = c( 1,  0, -1),
           "M - H" = c( 0,  1, -1))

warpbreaks.mc <- glht(warpbreaks.aov,
                      linfct = mcp(tension = K),
                      alternative = "less")

### correlation of first two tests is -1
cov2cor(vcov(warpbreaks.mc))

### use smallest of the two one-sided
### p-value as two-sided p-value -> 0.0232
summary(warpbreaks.mc)

### more complex models: Continuous outcome logistic
### regression; parameters are log-odds ratios
if (require("tram")) {
    confint(glht(Colr(breaks ~ wool + tension,
                      data = warpbreaks),
                 linfct = mcp("tension" = "Tukey")))
}
```

`glht-methods` *Methods for General Linear Hypotheses*

Description

Simultaneous tests and confidence intervals for general linear hypotheses.

Usage

```
## S3 method for class 'glht'
summary(object, test = adjusted(), ...)
```

```
## S3 method for class 'glht'
confint(object, parm, level = 0.95, calpha = adjusted_calpha(),
        ...)
## S3 method for class 'glht'
coef(object, rhs = FALSE, ...)
## S3 method for class 'glht'
vcov(object, ...)
## S3 method for class 'confint.glht'
plot(x, xlim, xlab, ylim, ...)
## S3 method for class 'glht'
plot(x, ...)
univariate()
adjusted(type = c("single-step", "Shaffer", "Westfall", "free",
         p.adjust.methods), ...)
Ftest()
Chisqtest()
adjusted_calpha(...)
univariate_calpha(...)
```

Arguments

object	an object of class glht.
test	a function for computing p values.
parm	additional parameters, currently ignored.
level	the confidence level required.
calpha	either a function computing the critical value or the critical value itself.
rhs	logical, indicating whether the linear function $K\hat{\theta}$ or the right hand side m (rhs = TRUE) of the linear hypothesis should be returned.
type	the multiplicity adjustment (adjusted) to be applied. See below and p.adjust.
x	an object of class glht or confint.glht.
xlim	the x limits (x1,x2) of the plot.
ylim	the y limits of the plot.
xlab	a label for the x axis.
...	additional arguments, such as maxpts, abseps or releps to pmvnorm in adjusted or qmvnorm in confint. Note that additional arguments specified to summary, confint, coef and vcov methods are currently ignored.

Details

The methods for general linear hypotheses as described by objects returned by glht can be used to actually test the global null hypothesis, each of the partial hypotheses and for simultaneous confidence intervals for the linear function $K\theta$.

The coef and vcov methods compute the linear function $K\hat{\theta}$ and its covariance, respectively.

The test argument to summary takes a function specifying the type of test to be applied. Classical Chisq (Wald test) or F statistics for testing the global hypothesis H_0 are implemented in functions

Chisqtest and Ftest. Several approaches to multiplicity adjusted p values for each of the linear hypotheses are implemented in function adjusted. The type argument to adjusted specifies the method to be applied: "single-step" implements adjusted p values based on the joint normal or t distribution of the linear function, and "Shaffer" and "Westfall" implement logically constraint multiplicity adjustments (Shaffer, 1986; Westfall, 1997). "free" implements multiple testing procedures under free combinations (Westfall et al, 1999). In addition, all adjustment methods implemented in p.adjust are available as well.

Simultaneous confidence intervals for linear functions can be computed using method confint. Univariate confidence intervals can be computed by specifying calpha = univariate_calpha() to confint. The critical value can directly be specified as a scalar to calpha as well. Note that plot(a) for some object a of class glht is equivalent to plot(confint(a)).

All simultaneous inference procedures implemented here control the family-wise error rate (FWER). Multivariate normal and t distributions, the latter one only for models of class lm, are evaluated using the procedures implemented in package mvtnorm. Note that the default procedure is stochastic. Reproducible p-values and confidence intervals require appropriate settings of seeds.

A more detailed description of the underlying methodology is available from Hothorn et al. (2008) and Bretz et al. (2010).

Value

summary computes (adjusted) p values for general linear hypotheses, confint computes (adjusted) confidence intervals. coef returns estimates of the linear function $K\theta$ and vcov its covariance.

References

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

Juliet P. Shaffer (1986), Modified sequentially rejective multiple test procedures. *Journal of the American Statistical Association*, **81**, 826–831.

Peter H. Westfall (1997), Multiple testing of general contrasts using logical constraints and correlations. *Journal of the American Statistical Association*, **92**, 299–306.

P. H. Westfall, R. D. Tobias, D. Rom, R. D. Wolfinger, Y. Hochberg (1999). *Multiple Comparisons and Multiple Tests Using the SAS System*. Cary, NC: SAS Institute Inc.

Torsten Hothorn, Frank Bretz and Peter Westfall (2008), Simultaneous Inference in General Parametric Models. *Biometrical Journal*, **50**(3), 346–363; See vignette("generalsiminf",package = "multcomp").

Examples

```
### set up a two-way ANOVA
amod <- aov(breaks ~ wool + tension, data = warpbreaks)

### set up all-pair comparisons for factor `tension'
wht <- glht(amod, linfct = mcp(tension = "Tukey"))

### 95% simultaneous confidence intervals
plot(print(confint(wht)))
```

```
### the same (for balanced designs only)
TukeyHSD(amod, "tension")

### corresponding adjusted p values
summary(wht)

### all means for levels of `tension'
amod <- aov(breaks ~ tension, data = warpbreaks)
glht(amod, linfct = matrix(c(1, 0, 0,
                             1, 1, 0,
                             1, 0, 1), byrow = TRUE, ncol = 3))

### confidence bands for a simple linear model, `cars' data
plot(cars, xlab = "Speed (mph)", ylab = "Stopping distance (ft)",
     las = 1)

### fit linear model and add regression line to plot
lmod <- lm(dist ~ speed, data = cars)
abline(lmod)

### a grid of speeds
speeds <- seq(from = min(cars$speed), to = max(cars$speed),
              length = 10)

### linear hypotheses: 10 selected points on the regression line != 0
K <- cbind(1, speeds)

### set up linear hypotheses
cht <- glht(lmod, linfct = K)

### confidence intervals, i.e., confidence bands, and add them plot
cci <- confint(cht)
lines(speeds, cci$confint[,"lwr"], col = "blue")
lines(speeds, cci$confint[,"upr"], col = "blue")


### simultaneous p values for parameters in a Cox model
if (require("survival") && require("MASS")) {
    data("leuk", package = "MASS")
    leuk.cox <- coxph(Surv(time) ~ ag + log(wbc), data = leuk)

    ### set up linear hypotheses
    lht <- glht(leuk.cox, linfct = diag(length(coef(leuk.cox))))

    ### adjusted p values
    print(summary(lht))
}
```

litter *Litter Weights Data Set*

Description

Dose response of litter weights in rats.

Usage

```
data("litter")
```

Format

This data frame contains the following variables

dose dosages at four levels: `0`, `5`, `50`, `500`.

gesttime gestation time as covariate.

number number of animals in litter as covariate.

weight response variable: average post-birth weights in the entire litter.

Details

Pregnant mice were divided into four groups and the compound in four different doses was administered during pregnancy. Their litters were evaluated for birth weights.

Source

P. H. Westfall, R. D. Tobias, D. Rom, R. D. Wolfinger, Y. Hochberg (1999). *Multiple Comparisons and Multiple Tests Using the SAS System*. Cary, NC: SAS Institute Inc., page 109.

P. H. Westfall (1997). Multiple Testing of General Contrasts Using Logical Constraints and Correlations. *Journal of the American Statistical Association*, **92**(437), 299–306.

Examples

```
### fit ANCOVA model to data
amod <- aov(weight ~ dose + gesttime + number, data = litter)

### define matrix of linear hypotheses for `dose'
doselev <- as.integer(levels(litter$dose))
K <- rbind(contrMat(table(litter$dose), "Tukey"),
           otrend = c(-1.5, -0.5, 0.5, 1.5),
           atrend = doselev - mean(doselev),
           ltrend = log(1:4) - mean(log(1:4)))

### set up multiple comparison object
Kht <- glht(amod, linfct = mcp(dose = K), alternative = "less")

### cf. Westfall (1997, Table 2)
summary(Kht, test = univariate())
summary(Kht, test = adjusted("bonferroni"))
summary(Kht, test = adjusted("Shaffer"))
summary(Kht, test = adjusted("Westfall"))
summary(Kht, test = adjusted("single-step"))
```

mmm *Simultaneous Inference for Multiple Marginal Models*

Description

Calculation of correlation between test statistics from multiple marginal models using the score decomposition

Usage

```
mmm(...)
mlf(...)
```

Arguments

`...`	A names argument list containing fitted models (`mmm`) or definitions of linear functions (`mlf`). If only one linear function is defined for `mlf`, it will be applied to all models in `mmm` by `glht.mlf`.

Details

Estimated correlations of the estimated parameters of interest from the multiple marginal models are obtained using a stacked version of the i.i.d. decomposition of parameter estimates by means of score components (first derivatives of the log likelihood). The method is less conservative than the Bonferroni correction. The details are provided by Pipper, Ritz and Bisgaard (2012).

The implementation assumes that the model were fitted to the same data, i.e., the rows of the matrices returned by `estfun` belong to the same observations for each model.

The reference distribution is always multivariate normal, if you want to use the multivariate t, please specify the corresponding degrees of freedom as an additional `df` argument to `glht`.

Observations with missing values contribute zero to the score function. Models have to be fitted using `na.exclude` as `na.action` argument.

Value

An object of class `mmm` or `mlf`, basically a named list of the arguments with a special method for `glht` being available for the latter. `vcov`, `estfun`, and `bread` methods are available for objects of class `mmm`.

Author(s)

Code for the computation of the joint covariance and sandwich matrices was contributed by Christian Ritz and Christian B. Pipper.

References

Christian Bressen Pipper, Christian Ritz and Hans Bisgaard (2012), A Versatile Method for Confirmatory Evaluation of the Effects of a Covariate in Multiple Models, *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, **61**, 315–326.

Examples

```
### replicate analysis of Hasler & Hothorn (2011),
### A Dunnett-Type Procedure for Multiple Endpoints,
### The International Journal of Biostatistics: Vol. 7: Iss. 1, Article 3.
### DOI: 10.2202/1557-4679.1258

### see ?coagulation
if (require("SimComp")) {
    data("coagulation", package = "SimComp")

    ### level "S" is the standard, "H" and "B" are novel procedures
    coagulation$Group <- relevel(coagulation$Group, ref = "S")

    ### fit marginal models
    (m1 <- lm(Thromb.count ~ Group, data = coagulation))
    (m2 <- lm(ADP ~ Group, data = coagulation))
    (m3 <- lm(TRAP ~ Group, data = coagulation))

    ### set-up Dunnett comparisons for H - S and B - S
    ### for all three models
    g <- glht(mmm(Thromb = m1, ADP = m2, TRAP = m3),
              mlf(mcp(Group = "Dunnett")), alternative = "greater")

    ### joint correlation
    cov2cor(vcov(g))

    ### simultaneous p-values adjusted by taking the correlation
    ### between the score contributions into account
    summary(g)
    ### simultaneous confidence intervals
    confint(g)

    ### compare with
    ## Not run:
        library("SimComp")
        SimCiDiff(data = coagulation, grp = "Group",
                  resp = c("Thromb.count","ADP","TRAP"),
                  type = "Dunnett", alternative = "greater",
                  covar.equal = TRUE)

## End(Not run)

    ### use sandwich variance matrix
    library("sandwich")
    g <- glht(mmm(Thromb = m1, ADP = m2, TRAP = m3),
              mlf(mcp(Group = "Dunnett")),
              alternative = "greater", vcov = sandwich)
    summary(g)
    confint(g)
}
```

```
### attitude towards science data
data("mn6.9", package = "TH.data")

### one model for each item
mn6.9.y1 <- glm(y1 ~ group, family = binomial(),
                na.action = na.omit, data = mn6.9)
mn6.9.y2 <- glm(y2 ~ group, family = binomial(),
                na.action = na.omit, data = mn6.9)
mn6.9.y3 <- glm(y3 ~ group, family = binomial(),
                na.action = na.omit, data = mn6.9)
mn6.9.y4 <- glm(y4 ~ group, family = binomial(),
                na.action = na.omit, data = mn6.9)

### test all parameters simulaneously
summary(glht(mmm(mn6.9.y1, mn6.9.y2, mn6.9.y3, mn6.9.y4),
             mlf(diag(2))))
### group differences
summary(glht(mmm(mn6.9.y1, mn6.9.y2, mn6.9.y3, mn6.9.y4),
             mlf("group2 = 0")))

### alternative analysis of Klingenberg & Satopaa (2013),
### Simultaneous Confidence Intervals for Comparing Margins of
### Multivariate Binary Data, CSDA, 64, 87-98
### http://dx.doi.org/10.1016/j.csda.2013.02.016

### see supplementary material for data description
### NOTE: this is not the real data but only a subsample
influenza <- structure(list(
HEADACHE = c(1L, 0L, 0L, 1L, 0L, 0L, 1L, 1L, 1L,
0L, 0L, 1L, 0L, 1L, 0L, 1L, 1L, 1L, 1L, 1L, 1L, 0L, 0L, 0L, 0L,
1L, 1L), MALAISE = c(0L, 0L, 1L, 1L, 0L, 1L, 1L, 1L, 0L, 1L,
0L, 0L, 1L, 1L, 0L, 0L, 1L, 0L, 1L, 0L, 1L, 0L, 1L, 1L, 0L, 1L,
0L), PYREXIA = c(0L, 0L, 0L, 0L, 0L, 1L, 0L, 1L, 0L, 0L, 1L,
1L, 1L, 1L, 0L, 0L, 0L, 0L, 0L, 1L, 1L, 1L, 0L, 0L, 0L, 1L, 1L
), ARTHRALGIA = c(0L, 0L, 0L, 0L, 1L, 0L, 1L, 0L, 1L, 1L, 0L,
0L, 1L, 1L, 0L, 1L, 0L, 0L, 1L, 0L, 0L, 0L, 0L, 1L, 1L, 1L, 1L
), group = structure(c(2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L,
2L, 2L, 2L, 2L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
1L), .Label = c("pla", "trt"), class = "factor"), Freq = c(32L,
165L, 10L, 23L, 3L, 1L, 4L, 2L, 4L, 2L, 1L, 1L, 1L, 1L, 167L,
1L, 11L, 37L, 7L, 7L, 5L, 3L, 3L, 1L, 2L, 4L, 2L)), .Names = c("HEADACHE",
"MALAISE", "PYREXIA", "ARTHRALGIA", "group", "Freq"), row.names = c(1L,
2L, 3L, 5L, 9L, 36L, 43L, 50L, 74L, 83L, 139L, 175L, 183L, 205L,
251L, 254L, 255L, 259L, 279L, 281L, 282L, 286L, 302L, 322L, 323L,
366L, 382L), class = "data.frame")
influenza <- influenza[rep(1:nrow(influenza), influenza$Freq), 1:5]

### Fitting marginal logistic regression models
(head_logreg <- glm(HEADACHE ~ group, data = influenza,
                    family = binomial()))
(mala_logreg <- glm(MALAISE ~ group, data = influenza,
                    family = binomial()))
(pyre_logreg <- glm(PYREXIA ~ group, data = influenza,
```

```
                    family = binomial()))
(arth_logreg <- glm(ARTHRALGIA ~ group, data = influenza,
                    family = binomial()))

### Simultaneous inference for log-odds
xy.sim <- glht(mmm(head = head_logreg,
                   mala = mala_logreg,
                   pyre = pyre_logreg,
                   arth = arth_logreg),
               mlf("grouptrt = 0"))
summary(xy.sim)
confint(xy.sim)

### Artificial examples
### Combining linear regression and logistic regression
set.seed(29)
y1 <- rnorm(100)
y2 <- factor(y1 + rnorm(100, sd = .1) > 0)
x1 <- gl(4, 25)
x2 <- runif(100, 0, 10)

m1 <- lm(y1 ~ x1 + x2)
m2 <- glm(y2 ~ x1 + x2, family = binomial())
### Note that the same explanatory variables are considered in both models
### but the resulting parameter estimates are on 2 different scales
### (original and log-odds scales)

### Simultaneous inference for the same parameter in the 2 model fits
summary(glht(mmm(m1 = m1, m2 = m2), mlf("x12 = 0")))

### Simultaneous inference for different parameters in the 2 model fits
summary(glht(mmm(m1 = m1, m2 = m2),
             mlf(m1 = "x12 = 0", m2 = "x13 = 0")))

### Simultaneous inference for different and identical parameters in the 2
### model fits
summary(glht(mmm(m1 = m1, m2 = m2),
             mlf(m1 = c("x12 = 0", "x13 = 0"), m2 = "x13 = 0")))

### Examples for binomial data
### Two independent outcomes
y1.1 <- rbinom(100, 1, 0.45)
y1.2 <- rbinom(100, 1, 0.55)
group <- factor(rep(c("A", "B"), 50))

m1 <- glm(y1.1 ~ group, family = binomial)
m2 <- glm(y1.2 ~ group, family = binomial)

summary(glht(mmm(m1 = m1, m2 = m2),
             mlf("groupB = 0")))

### Two perfectly correlated outcomes
y2.1 <- rbinom(100, 1, 0.45)
```

```
y2.2 <- y2.1
group <- factor(rep(c("A", "B"), 50))

m1 <- glm(y2.1 ~ group, family = binomial)
m2 <- glm(y2.2 ~ group, family = binomial)

summary(glht(mmm(m1 = m1, m2 = m2),
             mlf("groupB = 0")))

### use sandwich covariance matrix
summary(glht(mmm(m1 = m1, m2 = m2),
             mlf("groupB = 0"), vcov = sandwich))
```

---

modelparm *Generic Accessor Function for Model Parameters*

Description

Extract model parameters and their covariance matrix as well as degrees of freedom (if available) from a fitted model.

Usage

```
modelparm(model, coef., vcov., df, ...)
```

Arguments

model	a fitted model, for example an object returned by lm, glm, aov, survreg, or lmer etc.
coef.	an accessor function for the model parameters. Alternatively, the vector of coefficients.
vcov.	an accessor function for the covariance matrix of the model parameters. Alternatively, the covariance matrix directly.
df	an optional specification of the degrees of freedom to be used in subsequent computations.
...	additional arguments, currently ignored.

Details

One can't expect coef and vcov methods for arbitrary models to return a vector of p fixed effects model parameters (coef) and corresponding $p \times p$ covariance matrix (vcov).

The coef. and vcov. arguments can be used to define modified coef or vcov methods for a specific model. Methods for lmer and survreg objects are available (internally).

For objects inheriting from class lm the degrees of freedom are determined from model and the corresponding multivariate t distribution is used by all methods to glht objects. By default, the asymptotic multivariate normal distribution is used in all other cases unless df is specified by the user.

Value

An object of class `modelparm` with elements

- `coef` model parameters
- `vcov` covariance matrix of model parameters
- `df` degrees of freedom

`mtept` *Multiple Endpoints Data*

Description

Measurements on four endpoints in a two-arm clinical trial.

Usage

```
data(mtept)
```

Format

A data frame with 111 observations on the following 5 variables.

- `treatment` a factor with levels `Drug` `Placebo`
- `E1` endpoint 1
- `E2` endpoint 2
- `E3` endpoint 3
- `E4` endpoint 4

Details

The data (from Westfall et al., 1999) contain measurements of patients in treatment (`Drug`) and control (`Placebo`) groups, with four outcome variables.

Source

P. H. Westfall, R. D. Tobias, D. Rom, R. D. Wolfinger, Y. Hochberg (1999). *Multiple Comparisons and Multiple Tests Using the SAS System*. Cary, NC: SAS Institute Inc.

parm *Model Parameters*

Description

Directly specify estimated model parameters and their covariance matrix.

Usage

```
parm(coef, vcov, df = 0)
```

Arguments

coef	estimated coefficients.
vcov	estimated covariance matrix of the coefficients.
df	an optional specification of the degrees of freedom to be used in subsequent computations.

Details

When only estimated model parameters and the corresponding covariance matrix is available for simultaneous inference using glht (for example, when only the results but not the original data are available or, even worse, when the model has been fitted outside R), function parm sets up an object glht is able to compute on (mainly by offering coef and vcov methods).

Note that the linear function in glht can't be specified via mcp since the model terms are missing.

Value

An object of class parm with elements

coef	model parameters
vcov	covariance matrix of model parameters
df	degrees of freedom

Examples

```
## example from
## Bretz, Hothorn, and Westfall (2002).
## On multiple comparisons in R. R News, 2(3):14-17.

beta <- c(V1 = 14.8, V2 = 12.6667, V3 = 7.3333, V4 = 13.1333)
Sigma <- 6.7099 * (diag(1 / c(20, 3, 3, 15)))
confint(glht(model = parm(beta, Sigma, 37),
             linfct = c("V2 - V1 >= 0",
                        "V3 - V1 >= 0",
                        "V4 - V1 >= 0")),
        level = 0.9)
```

plot.cld *Plot a cld object*

Description

Plot information of `glht`, `summary.glht` or `confint.glht` objects stored as `cld` objects together with a compact letter display of all pair-wise comparisons.

Usage

```
## S3 method for class 'cld'
plot(x, type = c("response", "lp"), ...)
```

Arguments

`x`	An object of class `cld`.
`type`	Should the response or the linear predictor (lp) be plotted. If there are any covariates, the lp is automatically used. To use the response variable, set `type="response"` and `covar=FALSE` of the `cld` object.
`...`	Other optional print parameters which are passed to the plotting functions.

Details

This function plots the information stored in `glht`, `summary.glht` or `confint.glht` objects. Prior to plotting, these objects have to be converted to `cld` objects (see `cld` for details). All types of plots include a compact letter display (cld) of all pair-wise comparisons. Equal letters indicate no significant differences. Two levels are significantly different, in case they do not have any letters in common. If the fitted model contains any covariates, a boxplot of the linear predictor is generated with the cld within the upper margin. Otherwise, three different types of plots are used depending on the class of variable `y` of the `cld` object. In case of `class(y) == "numeric"`, a boxplot is generated using the response variable, classified according to the levels of the variable used for the Tukey contrast matrix. Is `class(y) == "factor"`, a mosaic plot is generated, and the cld is printed above. In case of `class(y) == "Surv"`, a plot of fitted survival functions is generated where the cld is plotted within the legend. The compact letter display is computed using the algorithm of Piepho (2004). Note: The user has to provide a sufficiently large upper margin which can be used to depict the compact letter display (see examples).

References

Hans-Peter Piepho (2004), An Algorithm for a Letter-Based Representation of All-Pairwise Comparisons, *Journal of Computational and Graphical Statistics*, **13**(2), 456–466.

See Also

`glht` `cld` `cld.summary.glht` `cld.confint.glht` `cld.glht` `boxplot` `mosaicplot` `plot.survfit`

Examples

```
### multiple comparison procedures
### set up a one-way ANOVA
data(warpbreaks)
amod <- aov(breaks ~ tension, data = warpbreaks)
### specify all pair-wise comparisons among levels of variable "tension"
tuk <- glht(amod, linfct = mcp(tension = "Tukey"))
### extract information
tuk.cld <- cld(tuk)
### use sufficiently large upper margin
old.par <- par(mai=c(1,1,1.25,1), no.readonly=TRUE)
### plot
plot(tuk.cld)
par(old.par)

### now using covariates
amod2 <- aov(breaks ~ tension + wool, data = warpbreaks)
tuk2 <- glht(amod2, linfct = mcp(tension = "Tukey"))
tuk.cld2 <- cld(tuk2)
old.par <- par(mai=c(1,1,1.25,1), no.readonly=TRUE)
### use different colors for boxes
plot(tuk.cld2, col=c("green", "red", "blue"))
par(old.par)

### get confidence intervals
ci.glht <- confint(tuk)
### plot them
plot(ci.glht)
old.par <- par(mai=c(1,1,1.25,1), no.readonly=TRUE)
### use 'confint.glht' object to plot all pair-wise comparisons
plot(cld(ci.glht), col=c("white", "blue", "green"))
par(old.par)

### set up all pair-wise comparisons for count data
data(Titanic)
mod <- glm(Survived ~ Class, data = as.data.frame(Titanic),
           weights = Freq, family = binomial())
### specify all pair-wise comparisons among levels of variable "Class"
glht.mod <- glht(mod, mcp(Class = "Tukey"))
### extract information
mod.cld <- cld(glht.mod)
### use sufficiently large upper margin
old.par <- par(mai=c(1,1,1.5,1), no.readonly=TRUE)
### plot
plot(mod.cld)
par(old.par)

### set up all pair-wise comparisons of a Cox-model
if (require("survival") && require("MASS")) {
  ### construct 4 classes of age
  Melanoma$Cage <- factor(sapply(Melanoma$age, function(x){
```

```
                              if( x <= 25 ) return(1)
                              if( x > 25 & x <= 50 ) return(2)
                              if( x > 50 & x <= 75 ) return(3)
                              if( x > 75 & x <= 100) return(4) }
                             ))
  ### fit Cox-model
  cm <- coxph(Surv(time, status == 1) ~ Cage, data = Melanoma)
  ### specify all pair-wise comparisons among levels of "Cage"
  cm.glht <- glht(cm, mcp(Cage = "Tukey"))
  # extract information & plot
  old.par <- par(no.readonly=TRUE)
  ### use mono font family
  if (dev.interactive())
      old.par <- par(family = "mono")
  plot(cld(cm.glht), col=c("black", "red", "blue", "green"))
  par(old.par)
}

if (require("nlme") && require("lme4")) {
  data("ergoStool", package = "nlme")

  stool.lmer <- lmer(effort ~ Type + (1 | Subject),
                     data = ergoStool)
  glme41 <- glht(stool.lmer, mcp(Type = "Tukey"))

  old.par <- par(mai=c(1,1,1.5,1), no.readonly=TRUE)
  plot(cld(glme41))
  par(old.par)
}
```

`recovery` *Recovery Time Data Set*

Description

Recovery time after surgery.

Usage

```
data("recovery")
```

Format

This data frame contains the following variables

blanket blanket type, a factor at four levels: `b0`, `b1`, `b2`, and `b3`.

minutes response variable: recovery time after a surgical procedure.

Details

A company developed specialized heating blankets designed to help the body heat following a surgical procedure. Four types of blankets were tried on surgical patients with the aim of comparing the recovery time of patients. One of the blanket was a standard blanket that had been in use already in various hospitals.

Source

P. H. Westfall, R. D. Tobias, D. Rom, R. D. Wolfinger, Y. Hochberg (1999). *Multiple Comparisons and Multiple Tests Using the SAS System*. Cary, NC: SAS Institute Inc., page 66.

Examples

```
### set up one-way ANOVA
amod <- aov(minutes ~ blanket, data = recovery)

### set up multiple comparisons: one-sided Dunnett contrasts
rht <- glht(amod, linfct = mcp(blanket = "Dunnett"),
            alternative = "less")

### cf. Westfall et al. (1999, p. 80)
confint(rht, level = 0.9)

### the same
rht <- glht(amod, linfct = mcp(blanket = c("b1 - b0 >= 0",
                                           "b2 - b0 >= 0",
                                           "b3 - b0 >= 0")))
confint(rht, level = 0.9)
```

---

sbp *Systolic Blood Pressure Data*

Description

Systolic blood pressure, age and gender of 69 people.

Usage

```
data("sbp")
```

Format

A data frame with 69 observations on the following 3 variables.

`gender` a factor with levels `male` `female`

`sbp` systolic blood pressure in mmHg

`age` age in years

Source

D. G. Kleinbaum, L. L. Kupper, K. E. Muller, A. Nizam, A. (1998), *Applied Regression Analysis and Other Multivariable Methods*, Duxbury Press, North Scituate, MA.

---

trees513 *Frankonian Tree Damage Data*

Description

Damages on young trees caused by deer browsing.

Usage

```
data("trees513")
```

Format

A data frame with 2700 observations on the following 4 variables.

`damage` a factor with levels `yes` and `no` indicating whether or not the trees has been damaged by game animals, mostly roe deer.

`species` a factor with levels `spruce`, `fir`, `pine`, `softwood (other)`, `beech`, `oak`, `ash/maple/elm/lime`, and `hardwood (other)`.

`lattice` a factor with levels `1`, ..., `53`, essentially a number indicating the position of the sampled area.

`plot` a factor with levels `x_1`, ..., `x_5` where `x` is the lattice. `plot` is nested within `lattice` and is a replication for each lattice point.

Details

In most parts of Germany, the natural or artificial regeneration of forests is difficult due to a high browsing intensity. Young trees suffer from browsing damage, mostly by roe and red deer. In order to estimate the browsing intensity for several tree species, the Bavarian State Ministry of Agriculture and Foresty conducts a survey every three years. Based on the estimated percentage of damaged trees, suggestions for the implementation or modification of deer management plans are made. The survey takes place in all 756 game management districts ('Hegegemeinschaften') in Bavaria. The data given here are from the game management district number 513 'Unterer Aischgrund' (located in Frankonia between Erlangen and H\"ochstadt) in 2006. The data of 2700 trees include the species and a binary variable indicating whether or not the tree suffers from damage caused by deer browsing.

Source

Bayerisches Staatsministerium fuer Landwirtschaft und Forsten (2006), Forstliche Gutachten zur Situation der Waldverjuengung 2006. https://www.stmelf.bayern.de/wald/

Torsten Hothorn, Frank Bretz and Peter Westfall (2008), Simultaneous Inference in General Parametric Models. *Biometrical Journal*, **50**(3), 346–363; See `vignette("generalsiminf",package = "multcomp")`.

Examples

```
summary(trees513)
```

---

waste *Industrial Waste Data Set*

Description

Industrial waste output in a manufactoring plant.

Usage

```
data("waste")
```

Format

This data frame contains the following variables

temp temperature, a factor at three levels: `low`, `medium`, `high`.

envir environment, a factor at five levels: `env1` ... `env5`.

waste response variable: waste output in a manufacturing plant.

Details

The data are from an experiment designed to study the effect of temperature (`temp`) and environment (`envir`) on waste output in a manufactoring plant. Two replicate measurements were taken at each temperature / environment combination.

Source

P. H. Westfall, R. D. Tobias, D. Rom, R. D. Wolfinger, Y. Hochberg (1999). *Multiple Comparisons and Multiple Tests Using the SAS System*. Cary, NC: SAS Institute Inc., page 177.

Examples

```
### set up two-way ANOVA with interactions
amod <- aov(waste ~ temp * envir, data=waste)

### comparisons of main effects only
K <- glht(amod, linfct = mcp(temp = "Tukey"))$linfct
K
glht(amod, K)

### comparisons of means (by averaging interaction effects)
low <- grep("low:envi", colnames(K))
```

```
med <- grep("medium:envi", colnames(K))
K[1, low] <- 1 / (length(low) + 1)
K[2, med] <- 1 / (length(low) + 1)
K[3, med] <- 1 / (length(low) + 1)
K[3, low] <- - 1 / (length(low) + 1)
K
confint(glht(amod, K))

### same as TukeyHSD
TukeyHSD(amod, "temp")

### set up linear hypotheses for all-pairs of both factors
wht <- glht(amod, linfct = mcp(temp = "Tukey", envir = "Tukey"))

### cf. Westfall et al. (1999, page 181)
summary(wht, test = adjusted("Shaffer"))
```

Index