

Package ‘multcomp’

September 10, 2009

Title Simultaneous Inference in General Parametric Models

Version 1.1-2

Date 2009-09-08

Author Torsten Hothorn, Frank Bretz and Peter Westfall with contributions by Richard M. Heiberger and Andre Schuetzenmeister

Description Simultaneous tests and confidence intervals for general linear hypotheses in parametric models, including linear, generalized linear, linear mixed effects, and survival models.

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

Depends stats, graphics, mvtnorm (>= 0.8-0), survival (>= 2.35-7)

Suggests lme4 (>= 0.999375-16), nlme, robustbase, mboost, coin, MASS, car, foreign, xtable

LazyData yes

License GPL-2

Repository CRAN

Date/Publication 2009-09-10 11:34:12

R topics documented:

cftest	2
cholesterol	3
cld	4
contrMat	6
detergent	7
glht	8
glht-methods	12
litter	15
modelparm	16
parm	17
plot.cld	18

recovery	20
trees513	22
waste	23

Index	25
--------------	-----------

cftest	<i>Testing Estimated Coefficients</i>
--------	---------------------------------------

Description

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

Usage

```
cftest(model, ...)
```

Arguments

model	a fitted model.
...	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

```
## The function is currently defined as
function(model, ...)
  summary(glht(model), test = univariate(), ...)

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, ...)
## S3 method for class 'glht':
cld(object, level = 0.05, ...)
## S3 method for class 'confint.glht':
cld(object, ...)

```

Arguments

<code>object</code>	An object of class <code>glht</code> , <code>summary.glht</code> or <code>confint.glht</code> .
<code>level</code>	Significance-level to be used to term a specific pair-wise comparison significant.
<code>...</code>	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:

<code>y</code>	Values of the response variable of the original model.
<code>yname</code>	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), AAn 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))
### 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))
### 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 = binom)
### 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))
### plott
plot(mod.cld)
par(old.par)
```

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, 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())
summary(dht, test = adjusted("Shaffer"))
summary(dht, test = adjusted("Westfall"))
```

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, ...)
mcp(..., interaction_average = FALSE, covariate_average = FALSE)
```

Arguments

<code>model</code>	a fitted model, for example an object returned by <code>lm</code> , <code>glm</code> , or <code>aov</code> etc. It is assumed that <code>coef</code> and <code>vcov</code> methods are available for <code>model</code> . For multiple comparisons of means, methods <code>model.matrix</code> , <code>model.frame</code> and <code>terms</code> are expected to be available for <code>model</code> as well.
<code>linfct</code>	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 <code>mcp</code> .
<code>alternative</code>	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.
<code>rhs</code>	an optional numeric vector specifying the right hand side of the hypothesis.
<code>interaction_average</code>	logical indicating if comparisons are averaging over interaction terms. Experimental!
<code>covariate_average</code>	logical indicating if comparisons are averaging over additional covariates. Experimental!
<code>...</code>	additional arguments to function <code>modelparm</code> in all <code>glht</code> methods. For function <code>mcp</code> , 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.

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).

Value

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

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

- 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)))

```

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 <code>glht</code> .
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 (<code>rhs = TRUE</code>) of the linear hypothesis should be returned.
type	the multiplicity adjustment (<code>adjusted</code>) to be applied. See below and <code>p.adjust</code> .
x	an object of class <code>glht</code> or <code>confint.glht</code> .
xlim	the x limits (<code>x1</code> , <code>x2</code>) 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`.

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

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

- 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"))

```

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 <code>lm</code> , <code>glm</code> , <code>aov</code> , <code>survreg</code> , or <code>lmer</code> etc.
coef.	an accessor function for the model parameters.
vcov.	an accessor function for the covariance matrix of the model parameters.
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

<code>coef</code>	model parameters
<code>vcov</code>	covariance matrix of model parameters
<code>df</code>	degrees of freedom

<code>parm</code>	<i>Model Parameters</i>
-------------------	-------------------------

Description

Directly specify estimated model parameters and their covariance matrix.

Usage

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

Arguments

<code>coef</code>	estimated coefficients.
<code>vcov</code>	estimated covariance matrix of the coefficients.
<code>df</code>	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

<code>coef</code>	model parameters
<code>vcov</code>	covariance matrix of model parameters
<code>df</code>	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

<code>x</code>	An object of class <code>cld</code> .
<code>type</code>	Should the response or the linear predictor (<code>lp</code>) be plotted. If there are any covariates, the <code>lp</code> is automatically used. To use the response variable, set <code>type="response"</code> and <code>covar=FALSE</code> of the <code>cld</code> object.
<code>...</code>	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))
### 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))
### 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))
### 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))
### 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()
  ### 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))
  plot(cld(glme41))
  par(old.par)
}

```

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)
```

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 Forestry 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 Höchstadt) 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. www.forst.bayern.de

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 manufacturing 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 manufacturing 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

*Topic **datasets**

- cholesterol, 3
- detergent, 7
- litter, 15
- recovery, 20
- trees513, 22
- waste, 23

*Topic **hplot**

- plot.cld, 18

*Topic **htest**

- cftest, 2
- glht, 8
- glht-methods, 12

*Topic **misc**

- contrMat, 6
- modelparm, 16
- parm, 17

- adjusted (*glht-methods*), 12
- adjusted_calpha (*glht-methods*), 12
- aov, 9, 16

- boxplot, 19

- cftest, 2
- Chisqtest (*glht-methods*), 12
- cholesterol, 3
- cld, 4, 18, 19
- cld.confint.glht, 19
- cld.confint.glht (*cld*), 4
- cld.glht, 19
- cld.glht (*cld*), 4
- cld.summary.glht, 19
- cld.summary.glht (*cld*), 4
- coef, 9, 10, 13, 16
- coef.glht (*glht-methods*), 12
- coeftest, 2
- confint, 10, 13
- confint.glht, 10
- confint.glht (*glht-methods*), 12

- contrMat, 6, 10

- detergent, 7

- Ftest (*glht-methods*), 12

- glht, 5, 8, 12, 13, 16, 17, 19
- glht-methods, 12
- glm, 9, 16

- litter, 15
- lm, 9, 13, 16

- mcp, 17
- mcp (*glht*), 8
- model.frame, 9
- model.matrix, 9
- modelparm, 9, 10, 16
- mosaicplot, 19

- p.adjust, 12, 13
- parm, 17
- plot.cld, 5, 18
- plot.confint.glht (*glht-methods*), 12
- plot.glht (*glht-methods*), 12
- plot.survfit, 19
- pmvnorm, 13

- qmvnorm, 13

- recovery, 20

- summary, 10
- summary.glht, 2, 10
- summary.glht (*glht-methods*), 12
- survreg, 16

- terms, 9
- trees513, 22

- univariate (*glht-methods*), 12

`univariate_calpha` (*glht-methods*),
12

`vcov`, 9, 10, 13, 16

`vcov.glht` (*glht-methods*), 12

`waste`, 23