Package 'goric'

July 22, 2025

Type Package

Title Generalized Order-Restricted Information Criterion

Version 1.1-3

Date 2025-04-01

Maintainer Daniel Gerhard <00gerhard@gmail.com>

Description Generalized Order-Restricted Information Criterion (GORIC) value for a set of hypotheses in multivariate linear models and generalised linear models.

Depends nlme

Imports quadprog, mvtnorm, MASS, Matrix, parallel

License GPL (>= 2)

LazyData yes

LazyLoad yes

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Daniel Gerhard [aut, cre],
Rebecca M. Kuiper [aut]

Repository CRAN

Date/Publication 2025-04-01 09:10:02 UTC

Contents

`constrMat` *Generate Constraint Matrices*

Description

Generate a constraint matrix with a predefined structure

Usage

```
constrMat(n, type = c("monotone", "control", "average", "laverage",
  "uaverage", "caverage"), base = 1)
```

Arguments

`n`	a (possibly named) vector of sample sizes for each group
`type`	character string defining the type of constraints; one of "monotone", "control","average","laverage","uaver or "caverage"
`base`	column of the constraint matrix representing a control group (when type = "control")

Value

a constraint matrix

See Also

`orlm`, `constrSet`

Examples

```
n <- c(10,20,30,40)
constrMat(n, type="monotone")
constrMat(n, type="control", base=2)
constrMat(n, type="average")
constrMat(n, type="laverage")
constrMat(n, type="uaverage")
constrMat(n, type="caverage", base=2)
```

constrSet *Generate Constraint Sets*

Description

Generate sets of constraint matrices (constr), right hand side elements, and numbers of equality constraints (nec) with a predefined structure

Usage

```
constrSet(n, set = c("sequence", "seqcontrol", "lplateau", "uplateau",
  "downturn", "williams"), direction = c("increase", "decrease"), base = 1)
```

Arguments

n	a (possibly named) vector of sample sizes for each group.
set	character string defining the type of constraints; one of "sequence", "seqcontrol", "lplateau", "uplateau", or "downturn"
direction	direction of the inequality constraints, either "increase" or "decrease"
base	column of the constraint matrix representing a control group

Value

a list with slots constr, rhs, and nec for each constraint definition

See Also

orlm, constrMat

Examples

```
n <- c(10,20,30,40)
constrSet(n, set="sequence")
constrSet(n, set="seqcontrol")
constrSet(n, set="lplateau")
constrSet(n, set="uplateau")
constrSet(n, set="downturn")
constrSet(n, set="williams")
```

goric *Calculate GORIC*

Description

The goric function calculates the order-restricted log likelihood, the penalty of the generalised order restricted information criterion (GORIC), the GORIC values, differences to the minimum GORIC value, and the GORIC weights for a set of hypotheses, where the penalty is based on $iter$ iterations. The hypothesis with the lowest GORIC value is the preferred one. The GORIC weights reflect the support of each hypothesis in the set. To compare two hypotheses (and not one to the whole set), one should examine the ratio of the two corresponding GORIC weights. To safequard for weak hypotheses (i.e., hypotheses not supported by the data), one should include a model with no constraints (the so-called unconstrained model).

Usage

```
goric(object, ..., iter = 1e+05, type = "GORIC", dispersion = 1,
  mc.cores = 1)

## S3 method for class 'orlm'
goric(object, ..., iter = 1e+05, type = "GORIC",
  mc.cores = 1)

## S3 method for class 'orgls'
goric(object, ..., iter = 1e+05, type = "GORIC",
  mc.cores = 1)

## S3 method for class 'list'
goric(object, ..., iter = 1e+05, type = "GORIC",
  dispersion = 1, mc.cores = 1)

## S3 method for class 'orglm'
goric(object, ..., iter = 1e+05, type = "GORIC",
  dispersion = 1, mc.cores = 1)
```

Arguments

object	an object of class orlm, orgls, orglm, or a list of these objects
...	further objects of class orlm, orgls, or orglm
iter	number of iterations to calculate GORIC penalty terms
type	if "GORIC" (default), the penalty term for the generalized order restriction information criterion is computed; with "GORICCa" or "GORICCb" small sample corrections for the penalty term are applied
dispersion	dispersion parameter to scale GORIC analogously to QAIC in generalized linear models
mc.cores	number of cores using a socket cluster implemented in package parallel

Value

a data.frame with the information criteria or a single penalty term

References

- Kuiper R.M., Hoijtink H., Silvapulle M.J. (2011). An Akaike-type Information Criterion for Model Selection Under Inequality Constraints. *Biometrika*, **98**, 495–501.
- Kuiper R.M., Hoijtink H., Silvapulle M.J. (2012). Generalization of the Order-Restricted Information Criterion for Multivariate Normal Linear Models. *Journal of Statistical Planning and Inference*, **142**, 2454-2463. doi:10.1016/j.jspi.2012.03.007.
- Kuiper R.M. and Hoijtink H. (submitted). A Fortran 90 Program for the Generalization of the Order-Restricted Information Criterion. Journal of Statictical Software.

See Also

orlm, orgls

Examples

```
## Example from Kuiper, R.M. and Hoijtink, H. (Unpublished).
# A Fortran 90 program for the generalization of the
# order restricted information criterion.
# constraint definition
cmat <- cbind(diag(3), 0) + cbind(0, -diag(3))
constr <- kronecker(diag(3), cmat)
constr

# no effect model
(fm0 <- orlm(cbind(SDH, SGOT, SGPT) ~ dose-1, data=vinylidene,
            constr=constr, rhs=rep(0, nrow(constr)), nec=nrow(constr)))

# order constrained model (increasing serum levels with increasing doses)
fm1 <- orlm(cbind(SDH, SGOT, SGPT) ~ dose-1, data=vinylidene,
            constr=constr, rhs=rep(0, nrow(constr)), nec=0)
summary(fm1)

# unconstrained model
(fmunc <- orlm(cbind(SDH, SGOT, SGPT) ~ dose-1, data=vinylidene,
              constr=matrix(0, nrow=1, ncol=12), rhs=0, nec=0))

# calculate GORIC
# (only small number of iterations to decrease computation time, default: iter=100000)
goric(fm0, fm1, fmunc, iter=1000)
```

goric_penalty *GORIC penalty term*

Description

Calculates the GORIC penalty term (level probability) by Monte-Carlo simulation.

Usage

```
goric_penalty(object, iter = 1e+05, type = "GORIC", mc.cores = 1)

orglm_penalty(object, iter = 1e+05, type = "GORIC", mc.cores = 1)
```

Arguments

object	an object of class orlm, orgls (or orglm for function orglm_penalty)
iter	number of iterations to calculate GORIC penalty terms
type	if "GORIC" (default), the penalty term for the generalized order restriction information criterion is computed; with "GORICCa" or "GORICCb" small sample corrections for the penalty term are applied
mc.cores	number of cores using a socket cluster implemented in package parallel

See Also

orlm, orgls, orglm

orglm *Fitting Order-Restricted Generalised Linear Models*

Description

orglm is used to fit generalised linear models with restrictions on the parameters, specified by giving a description of the linear predictor, a description of the error distribution, and a description of a matrix with linear constraints. The quadprog package is used to apply linear constraints on the parameter vector.

Usage

```
orglm(formula, family = gaussian, data, weights, subset, na.action,
  start = NULL, etastart, mustart, offset, control = list(...),
  model = TRUE, method = "orglm.fit", x = FALSE, y = TRUE,
  contrasts = NULL, constr, rhs, nec, ...)

orglm.fit(x, y, weights = rep(1, nobs), start = NULL, etastart = NULL,
  mustart = NULL, offset = rep(0, nobs), family = gaussian(),
  control = list(), intercept = TRUE, constr, rhs, nec)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
family	a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See family for details of family functions.)
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which orglm is called.
weights	an optional vector of 'prior weights' to be used in the fitting process. Should be NULL or a numeric vector.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful.
start	starting values for the parameters in the linear predictor.
etastart	starting values for the linear predictor.
mustart	starting values for the vector of means.
offset	this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases. One or more offset terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See model.offset.
control	a list of parameters for controlling the fitting process. For orglm this is passed to glm.control.
model	a logical value indicating whether *model frame* should be included as a component of the returned value.
method	the method to be used in fitting the model. The default method "orglm.fit" uses iteratively reweighted least squares with a quadradic programming step included at each iteration.
x	is a design matrix of dimension n * p
y	is a vector of observations of length n.
contrasts	an optional list. See the contrasts.arg of model.matrix.default.
constr	a matrix with linear constraints. The columns of this matrix should correspond to the columns of the design matrix.
rhs	right hand side of the linear constraint formulation. A numeric vector with a length corresponding to the rows of constr.
nec	Number of equality constrints. The first nec constraints defined in constr are treated as equality constraints; the remaining ones are inequality constraints.

`...`	For `orglm`: arguments to be used to form the default `control` argument if it is not supplied directly.
`intercept`	logical. Should an intercept be included in the *null* model?

Details

Non-`NULL` `weights` can be used to indicate that different observations have different dispersions (with the values in `weights` being inversely proportional to the dispersions); or equivalently, when the elements of `weights` are positive integers w_i, that each response y_i is the mean of w_i unit-weight observations. For a binomial GLM prior weights are used to give the number of trials when the response is the proportion of successes: they would rarely be used for a Poisson GLM. If more than one of `etastart`, `start` and `mustart` is specified, the first in the list will be used. It is often advisable to supply starting values for a `quasi` family, and also for families with unusual links such as `gaussian("log")`. For the background to warning messages about 'fitted probabilities numerically 0 or 1 occurred' for binomial GLMs, see Venables & Ripley (2002, pp. 197–8).

Value

An object of class `"orglm"` is a list containing at least the following components:

coefficients a named vector of coefficients

residuals the *working* residuals, that is the residuals in the final iteration of the IWLS fit. Since cases with zero weights are omitted, their working residuals are `NA`.

fitted.values the fitted mean values, obtained by transforming the linear predictors by the inverse of the link function.

rank the numeric rank of the fitted linear model.

family the `family` object used.

linear.predictors the linear fit on link scale.

deviance up to a constant, minus twice the maximized log-likelihood. Where sensible, the constant is chosen so that a saturated model has deviance zero.

null.deviance The deviance for the null model, comparable with `deviance`. The null model will include the offset, and an intercept if there is one in the model. Note that this will be incorrect if the link function depends on the data other than through the fitted mean: specify a zero offset to force a correct calculation.

iter the number of iterations of IWLS used.

weights the *working* weights, that is the weights in the final iteration of the IWLS fit.

prior.weights the weights initially supplied, a vector of 1s if none were.

df.residual the residual degrees of freedom of the unconstrained model.

df.null the residual degrees of freedom for the null model.

y if requested (the default) the y vector used. (It is a vector even for a binomial model.)

converged logical. Was the IWLS algorithm judged to have converged?

boundary logical. Is the fitted value on the boundary of the attainable values?

Author(s)

Modification of the original glm.fit by Daniel Gerhard. The original R implementation of `glm` was written by Simon Davies working for Ross Ihaka at the University of Auckland, but has since been extensively re-written by members of the R Core team. The design was inspired by the S function of the same name described in Hastie & Pregibon (1992).

References

- Dobson, A. J. (1990) *An Introduction to Generalized Linear Models.* London: Chapman and Hall.
- Hastie, T. J. and Pregibon, D. (1992) *Generalized linear models.* Chapter 6 of *Statistical Models in S* eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks//Cole.
- McCullagh P. and Nelder, J. A. (1989) *Generalized Linear Models.* London: Chapman and Hall.
- Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S.* New York: Springer.

See Also

`glm`, `solve.QP`

`orgls` *Fitting generalized least squares regression models with order restrictions*

Description

`orgls` is used to fit generalised least square models analogously to the function `gls` in package `nlme` but with order restrictions on the parameters.

Usage

```
orgls(formula, data, constr, rhs, nec, weights = NULL, correlation = NULL,
  control = orlmcontrol())

## S3 method for class 'formula'
orgls(formula, data, constr, rhs, nec, weights = NULL,
  correlation = NULL, control = orlmcontrol())
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which orgls is called.

constr	matrix with constraints; with rows as constraint definition, columns should be in line with the parameters of the model
rhs	vector of right hand side elements; $Constr\ \theta \geq rhs$; number should equal the number of rows of the constr matrix
nec	number of equality constraints; a numeric value treating the first nec constr rows as equality constraints, or a logical vector with TRUE for equality- and FALSE for inequality constraints.
weights	a varClasses object; more details are provided on the help pages in R package nlme
correlation	a corClasses object; more details are provided on the help pages in R package nlme
control	a list of control arguments; see orlmcontrol for details.

Details

The contraints in the hypothesis of interest are defined by $constr$, rhs, and nec. The first nec constraints are the equality contraints: $Constr[1:nec, 1:tk]\theta = rhs[1:nec]$; and the remaing ones are the inequality contraints: $Constr[nec+1:c_m, 1:tk]\theta \geq rhs[nec+1:c_m]$. Two requirements should be met:

1. The first nec constraints must be the equality contraints (i.e., $Constr[1:nec, 1:tk]\theta = rhs[1:nec]$) and the remaining ones the inequality contraints (i.e., $Constr[nec+1:c_m, 1:tk]\theta \geq rhs[nec+1:c_m]$).
2. When rhs is not zero, $Constr$ should be of full rank (after discarding redundant restrictions).

Value

an object of class orgls

References

- Kuiper R.M., Hoijtink H., Silvapulle M.J. (2011). An Akaike-type Information Criterion for Model Selection Under Inequality Constraints. *Biometrika*, **98**, 495–501.
- Kuiper R.M., Hoijtink H., Silvapulle M.J. (2012). Generalization of the Order-Restricted Information Criterion for Multivariate Normal Linear Models. *Journal of Statistical Planning and Inference*, **142**, 2454-2463. doi:10.1016//j.jspi.2012.03.007.
- Kuiper R.M. and Hoijtink H. (submitted). A Fortran 90 Program for the Generalization of the Order-Restricted Information Criterion. Journal of Statictical Software.

See Also

solve.QP, goric

Examples

```
# generating example data
library(mvtnorm)
# group means
m <- c(0,5,5,7)
# compound symmetry structure of residuals
# (10 individuals per group, rho=0.7)
cormat <- kronecker(diag(length(m)*10), matrix(0.7, nrow=length(m), ncol=length(m)))
diag(cormat) <- 1
# different variances per group
sds <- rep(c(1,2,0.5,1), times=10*length(m))
sigma <- crossprod(diag(sds), crossprod(cormat, diag(sds)))
response <- as.vector(rmvnorm(1, rep(m, times=10*length(m)), sigma=sigma))
dat <- data.frame(response,
                  grp=rep(LETTERS[1:length(m)], times=10*length(m)),
                  ID=as.factor(rep(1:(10*length(m)), each=length(m))))

## set of gls models:
# unconstrained model
m1 <- orgls(response ~ grp-1, data = dat,
            constr=rbind(c(0,0,0,0)), rhs=0, nec=0,
            weights=varIdent(form=~1|grp),
            correlation=corCompSymm(form=~1|ID))

# simple order
m2 <- orgls(response ~ grp-1, data = dat,
            constr=rbind(c(-1,1,0,0),c(0,-1,1,0),c(0,0,-1,1)), rhs=c(0,0,0), nec=0,
            weights=varIdent(form=~1|grp),
            correlation=corCompSymm(form=~1|ID))

# equality constraints
m3 <- orgls(response ~ grp-1, data = dat,
            constr=rbind(c(-1,1,0,0),c(0,-1,1,0),c(0,0,-1,1)), rhs=c(0,0,0), nec=3,
            weights=varIdent(form=~1|grp),
            correlation=corCompSymm(form=~1|ID))
```

`orglsSet` *Set of generalised least-squares models*

Description

Fitting a specific set of generalisd least-squares models with order restrictions.

Usage

```
orglsSet(formula, data, weights = NULL, correlation = NULL, set,
  direction = "increase", n = NULL, base = 1, control = orlmcontrol())
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
weights	a varClasses object; more details are provided on the help pages in R package nlme
correlation	a corClasses object; more details are provided on the help pages in R package nlme
set	either a character string (see constrSet), or a list with slots for constr, rhs, and nec similarly defined as in orlm
direction	direction of the order constraints
n	a (possibly named) vector of sample sizes for each group
base	column of the constraint matrix representing a control group
control	a list of control arguments; see orlmcontrol for details.

Details

This function is just a wrapper for repeated calls of orgls with different constraint definitions. Predefined lists with constraint-sets can be constructed with function constrSet.

Value

a list with orgls objects

See Also

orgls, constrSet, goric

---

orlm *Fitting multivariate regression models with order restrictions*

Description

This is a modification of the lm function, fitting (multivariate) linear models with order constraints on the model coefficients.

Usage

```
orlm(formula, data, constr, rhs, nec, control = orlmcontrol())

## S3 method for class 'formula'
orlm(formula, data, constr, rhs, nec,
  control = orlmcontrol())
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
constr	matrix with constraints; with rows as constraint definition, columns should be in line with the parameters of the model
rhs	vector of right hand side elements; $Constr\ \theta \geq rhs$; number should equal the number of rows of the constr matrix
nec	number of equality constraints; a numeric value treating the first nec constr rows as equality constraints, or a logical vector with TRUE for equality- and FALSE for inequality constraints.
control	a list of control arguments; see orlmcontrol for details.

Details

The contraints in the hypothesis of interest are defined by $Constr$, rhs, and nec. The first nec constraints are the equality contraints: $Constr[1:nec, 1:tk]\theta = rhs[1:nec]$; and the remaing ones are the inequality contraints: $Constr[nec+1:c_m, 1:tk]\theta \geq rhs[nec+1:c_m]$. Two requirements should be met:

1. The first nec constraints must be the equality contraints (i.e., $Constr[1:nec, 1:tk]\theta = rhs[1:nec]$) and the remaining ones the inequality contraints (i.e., $Constr[nec+1:c_m, 1:tk]\theta \geq rhs[nec+1:c_m]$).
2. When rhs is not zero, $Constr$ should be of full rank (after discarding redundant restrictions).

Value

an object of class orlm

References

- Kuiper R.M., Hoijtink H., Silvapulle M.J. (2011). An Akaike-type Information Criterion for Model Selection Under Inequality Constraints. *Biometrika*, **98**, 495–501.
- Kuiper R.M., Hoijtink H., Silvapulle M.J. (2012). Generalization of the Order-Restricted Information Criterion for Multivariate Normal Linear Models. *Journal of Statistical Planning and Inference*, **142**, 2454-2463. doi:10.1016//j.jspi.2012.03.007.
- Kuiper R.M. and Hoijtink H. (submitted). A Fortran 90 Program for the Generalization of the Order-Restricted Information Criterion. Journal of Statictical Software.

See Also

solve.QP, goric

Examples

```
########################
## Artificial example ##
########################
n <- 10
m <- c(1,2,1,5)
nm <- length(m)
dat <- data.frame(grp=as.factor(rep(1:nm, each=n)),
                  y=rnorm(n*nm, rep(m, each=n), 1))

# unrestricted linear model
cm1 <- matrix(0, nrow=1, ncol=4)
fm1 <- orlm(y ~ grp-1, data=dat, constr=cm1, rhs=0, nec=0)

# order restriction (increasing means)
cm2 <- rbind(c(-1,1,0,0),
             c(0,-1,1,0),
             c(0,0,-1,1))
fm2 <- orlm(y ~ grp-1, data=dat, constr=cm2,
            rhs=rep(0,nrow(cm2)), nec=0)

# order restriction (increasing at least by delta=1)
fm3 <- orlm(y ~ grp-1, data=dat, constr=cm2,
            rhs=rep(1,nrow(cm2)), nec=0)

# larger than average of the neighboring first 2 parameters
cm4 <- rbind(c(-0.5,-0.5,1,0),
             c(0,-0.5,-0.5,1))
fm4 <- orlm(y ~ grp-1, data=dat, constr=cm4,
            rhs=rep(0,nrow(cm4)), nec=0)

# equality constraints (all parameters equal)
fm5 <- orlm(y ~ grp-1, data=dat, constr=cm2,
            rhs=rep(0,nrow(cm2)), nec=nrow(cm2))

# alternatively
fm5 <- orlm(y ~ grp-1, data=dat, constr=cm2,
            rhs=rep(0,nrow(cm2)), nec=c(TRUE,TRUE,TRUE))

# constraining the 1st and the 4th parameter
# to their true values, and the 2nd and 3rd between them
cm6 <- rbind(c( 1,0,0,0),
             c(-1,1,0,0),
             c(0,-1,0,1),
             c(-1,0,1,0),
             c(0,0,-1,1),
             c(0,0, 0,1))
fm6 <- orlm(y ~ grp-1, data=dat, constr=cm6,
            rhs=c(1,rep(0,4),5), nec=c(TRUE,rep(FALSE,4),TRUE))


################################################################
```

```
## Example from Kuiper, R.M. and Hoijtink, H. (Unpublished). ##
## A Fortran 90 program for the generalization of the        ##
## order restricted information criterion.                   ##
###############################################################

# constraint definition
cmat <- cbind(diag(3), 0) + cbind(0, -diag(3))
constr <- kronecker(diag(3), cmat)

# no effect model
(fm0 <- orlm(cbind(SDH, SGOT, SGPT) ~ dose-1, data=vinylidene,
             constr=constr, rhs=rep(0, nrow(constr)), nec=nrow(constr)))

# order constrained model (increasing serum levels with increasing doses)
fm1 <- orlm(cbind(SDH, SGOT, SGPT) ~ dose-1, data=vinylidene,
            constr=constr, rhs=rep(0, nrow(constr)), nec=0)
summary(fm1)

# unconstrained model
(fmunc <- orlm(cbind(SDH, SGOT, SGPT) ~ dose-1, data=vinylidene,
               constr=matrix(0, nrow=1, ncol=12), rhs=0, nec=0))
```

`orlmcontrol` *Control arguments for the orlm function.*

Description

A list with control arguments controlling the orlm function

Usage

```
orlmcontrol(maxiter = 10000, absval = 1e-04)
```

Arguments

`maxiter`	maximum number of iterations
`absval`	tolerance criterion for convergence

Value

a list with control arguments

See Also

`orlm`

`orlmSet` *Set of multivariate regression models*

Description

Fitting a specific set of multivariate regression models with order restrictions.

Usage

```
orlmSet(formula, data, set, direction = "increase", n = NULL, base = 1,
  control = orlmcontrol())
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
`set`	either a character string (see `constrSet`), or a list with slots for constr, rhs, and nec similarly defined as in `orlm`
`direction`	direction of the order constraints
`n`	a (possibly named) vector of sample sizes for each group
`base`	column of the constraint matrix representing a control group
`control`	a list of control arguments; see `orlmcontrol` for details.

Details

This function is just a wrapper for repeated calls of `orlm` with different constraint definitions. Predefined lists with constraint-sets can be constructed with function `constrSet`.

Value

a list with orlm objects

See Also

`orlm`, `constrSet`, `goric`

Examples

```
########################
## Artificial example ##
########################

n <- 10
m <- c(1,2,4,5,2,1)
nm <- length(m)
dat <- data.frame(grp=as.factor(rep(1:nm, each=n)),
                  y=rnorm(n*nm, rep(m, each=n), 1))

(cs <- constrSet(table(dat$grp), set="sequence"))
(oss <- orlmSet(y ~ grp-1, data=dat, set=cs))

# the same as:
oss <- orlmSet(y ~ grp-1, data=dat, set="sequence")
```

---

sim *Simulation from order restricted linear models*

Description

Simulation function for orlm and orgls objects

Usage

```
sim(object, n.sims)

## S3 method for class 'orlm'
sim(object, n.sims)

## S3 method for class 'orgls'
sim(object, n.sims)
```

Arguments

object	an object of class "orlm" or "orgls".
n.sims	number of simulation replications.

Details

Given the estimated coefficients of a orlm or orgls model, a set new parameters are generated. n.sims new sets of observations are generated based on the unrestricted model; these new datasets are used to estimate a new set of model coefficients incorporating the given order restrictions.

Value

a list with sets of simulated parameters.

See Also

orlm, orgls

Examples

```
########################
## Artificial example ##
########################
n <- 10
m <- c(1,1,2)
dat <- data.frame(grp=as.factor(rep(1:length(m), each=n)),
                  y=rnorm(n*length(m), rep(m, each=n), 1))
cm <- rbind(c(-1,1,0),
            c(0,-1,1))
fm <- orlm(y ~ grp-1, data=dat, constr=cm, rhs=rep(0,nrow(cm)), nec=0)
b <- sim(fm, n.sims=1000)$coef
pairs(t(b), cex=0.3)
```

---

vinylidene *Effect of vinylidene fluoride on liver cancer*

Description

Real data which are available on page 10 of Silvapulle and Sen (2005) and in a report prepared by Litton Bionetics Inc in 1984. These data were used in an experiment to find out whether vinylidene fluoride gives rise to liver damage. Since increased levels of serum enzyme are inherent in liver damage, the focus is on whether enzyme levels are affected by vinylidene fluoride. The variable of interest is the serum enzyme level. Three types of enzymes are inspected, namely SDH, SGOT, and SGPT. To study whether vinylidene fluoride has an influence on the three serum enzymes, four dosages of this substance are examined. In each of these four treatment groups, ten male Fischer-344 rats received the substance.

Usage

```
vinylidene
```

Format

A data frame with 40 observations on the following 4 variables.

`SDH` serum enzyme level of enzyme type SDH.

`SGOT` serum enzyme level of enzyme type SGOT.

`SGPT` serum enzyme level of enzyme type SGPT.

`dose` factor with 4 levels (d1-d4) representing the 4 vinylidene fluoride concentrations.

References

Silvapulle MJ, Sen PK (2005). *Constrained Statistical Inference*. New Jersey: Wiley.

Index