Package ‘subselect’

July 23, 2025

Version 0.16.0

Date 2025-03-29

Title Selecting Variable Subsets

Author Jorge Orestes Cerdeira [aut],
Pedro Duarte Silva [aut, cre],
Jorge Cadima [aut],
Manuel Minhoto [aut]

Maintainer Pedro Duarte Silva <psilva@ucp.pt>

Description A collection of functions which (i) assess the quality of variable subsets as surrogates for a full data set, in either an exploratory data analysis or in the context of a multivariate linear model, and (ii) search for subsets which are optimal under various criteria. Theoretical support for the heuristic search methods and exploratory data analysis criteria is in Cadima, Cerdeira, Minhoto (2003, <doi:10.1016/j.csda.2003.11.001>). Theoretical support for the leap and bounds algorithm and the criteria for the general multivariate linear model is in Duarte Silva (2001, <doi:10.1006/jmva.2000.1920>). There is a package vignette ``subselect'', which includes additional references.

Depends R (>= 3.5.0)

Imports MASS, ISwR, corpcor

License GPL (>= 2)

Repository CRAN

LazyData TRUE

NeedsCompilation yes

Date/Publication 2025-03-31 17:10:01 UTC

Contents

`anneal` *Simulated Annealing Search for an optimal k-variable subset*

Description

Given a set of variables, a Simulated Annealing algorithm seeks a k-variable subset which is optimal, as a surrogate for the whole set, with respect to a given criterion.

Usage

```
anneal( mat, kmin, kmax = kmin, nsol = 1, niter = 1000, exclude
= NULL, include = NULL, improvement = TRUE, setseed = FALSE,
cooling = 0.05,  temp = 1, coolfreq = 1, criterion = "default",
pcindices = "first_k", initialsol=NULL, force=FALSE, H=NULL, r=0,
tolval=1000*.Machine$double.eps,tolsym=1000*.Machine$double.eps)
```

Arguments

`mat`	a covariance/correlation, information or sums of squares and products matrix of the variables from which the k-subset is to be selected. See the `Details` section below.
`kmin`	the cardinality of the smallest subset that is wanted.
`kmax`	the cardinality of the largest subset that is wanted.
`nsol`	the number of initial/final subsets (runs of the algorithm).
`niter`	the number of iterations of the algorithm for each initial subset.
`exclude`	a vector of variables (referenced by their row/column numbers in matrix `mat`) that are to be forcibly excluded from the subsets.
`include`	a vector of variables (referenced by their row/column numbers in matrix `mat`) that are to be forcibly included in the subsets.

`improvement` a logical variable indicating whether or not the best final subset (for each cardinality) is to be passed as input to a local improvement algorithm (see function `improve`).

`setseed` logical variable indicating whether to fix an initial seed for the random number generator, which will be re-used in future calls to this function whenever setseed is again set to TRUE.

`cooling` variable in the]0,1[interval indicating the rate of geometric cooling for the Simulated Annealing algorithm.

`temp` positive variable indicating the initial temperature for the Simulated Annealing algorithm.

`coolfreq` positive integer indicating the number of iterations of the algorithm between coolings of the temperature. By default, the temperature is cooled at every iteration.

`criterion` Character variable, which indicates which criterion is to be used in judging the quality of the subsets. Currently, the "RM", "RV", "GCD", "Tau2", "Xi2", "Zeta2", "ccr12" and "Wald" criteria are supported (see the `Details` section, the `References` and the links `rm.coef`, `rv.coef`, `gcd.coef`, `tau2.coef`, `xi2.coef`, `zeta2.coef` and `ccr12.coef` for further details). The default criterion is "Rm" if parameter `r` is zero (exploratory and PCA problems), "Wald" if `r` is equal to one and `mat` has a "FisherI" attribute set to TRUE (generalized linear models), and "Tau2" otherwise (multivariate linear model framework).

`pcindices` either a vector of ranks of Principal Components that are to be used for comparison with the k-variable subsets (for the GCD criterion only, see `gcd.coef`) or the default text `first_k`. The latter will associate PCs 1 to *k* with each cardinality *k* that has been requested by the user.

`initialsol` vector, matrix or 3-d array of initial solutions for the simulated annealing search. If a *single cardinality* is required, `initialsol` may be a vector of length *k*, in which case it is used as the initial solution for all `nsol` final solutions that are requested; a 1 x *k* matrix (as produced by the `$bestsets` output value of the algorithm functions `anneal`, `genetic`, or `improve`), or a 1 x *k* x 1 array (as produced by the `$subsets` output value), in which case it will be treated as the above k-vector; or an `nsol` x `k` matrix, or `nsol` x `k` x 1 3-d array, in which case each row (dimension 1) will be used as the initial solution for each of the `nsol` final solutions requested. If *more than one cardinality* is requested, `initialsol` can be a `length(kmin:kmax)` x `kmax` matrix (as produced by the `$bestsets` option of the algorithm functions), in which case each row will be replicated to produced the initial solution for all `nsol` final solutions requested in each cardinality, or a `nsol` x `kmax` x `length(kmin:kmax)` 3-d array (as produced by the `$subsets` output option), in which case each row (dimension 1) is interpreted as a different initial solution.

If the `exclude` and/or `include` options are used, `initialsol` must also respect those requirements.

`force` a logical variable indicating whether, for large data sets (currently `p` > 400) the algorithm should proceed anyways, regardless of possible memory problems which may crash the R session.

H — Effect description matrix. Not used with the RM, RV or GCD criteria, hence the NULL default value. See the `Details` section below.

r — Expected rank of the effects (`H`) matrix. Not used with the RM, RV or GCD criteria. See the `Details` section below.

tolval — the tolerance level for the reciprocal of the 2-norm condition number of the correlation/covariance matrix, i.e., for the ratio of the smallest to the largest eigenvalue of the input matrix. Matrices with a reciprocal of the condition number smaller than `tolval` will activate a restricted-search for well conditioned subsets.

tolsym — the tolerance level for symmetry of the covariance/correlation/total matrix and for the effects (`H`) matrix. If corresponding matrix entries differ by more than this value, the input matrices will be considered asymmetric and execution will be aborted. If corresponding entries are different, but by less than this value, the input matrix will be replaced by its symmetric part, i.e., input matrix A becomes (A+t(A))/2.

Details

An initial k-variable subset (for k ranging from `kmin` to `kmax`) of a full set of p variables is randomly selected and passed on to a Simulated Annealing algorithm. The algorithm then selects a random subset in the neighbourhood of the current subset (neighbourhood of a subset S being defined as the family of all k-variable subsets which differ from S by a single variable), and decides whether to replace the current subset according to the Simulated Annealing rule, i.e., either (i) always, if the alternative subset's value of the criterion is higher; or (ii) with probability $\exp^{\frac{ac-cc}{t}}$ if the alternative subset's value of the criterion (ac) is lower than that of the current solution (cc), where the parameter t (temperature) decreases throughout the iterations of the algorithm. For each cardinality k, the stopping criterion for the algorithm is the number of iterations (`niter`) which is controlled by the user. Also controlled by the user are the initial temperature (`temp`) the rate of geometric cooling of the temperature (`cooling`) and the frequency with which the temperature is cooled, as measured by `coolfreq`, the number of iterations after which the temperature is multiplied by 1-`cooling`.

Optionally, the best k-variable subset produced by Simulated Annealing may be passed as input to a restricted local search algorithm, for possible further improvement.

The user may force variables to be included and/or excluded from the k-subsets, and may specify initial solutions.

For each cardinality k, the total number of calls to the procedure which computes the criterion values is `nsol` x (`niter` + 1). These calls are the dominant computational effort in each iteration of the algorithm.

In order to improve computation times, the bulk of computations is carried out by a Fortran routine. Further details about the Simulated Annealing algorithm can be found in Reference 1 and in the comments to the Fortran code (in the `src` subdirectory for this package). For datasets with a very large number of variables (currently p > 400), it is necessary to set the `force` argument to TRUE for the function to run, but this may cause a session crash if there is not enough memory available.

The function checks for ill-conditioning of the input matrix (specifically, it checks whether the ratio of the input matrix's smallest and largest eigenvalues is less than `tolval`). For an ill-conditioned input matrix, the search is restricted to its well-conditioned subsets. The function `trim.matrix` may be used to obtain a well-conditioned input matrix.

In a general descriptive (Principal Components Analysis) setting, the three criteria Rm, Rv and Gcd can be used to select good k-variable subsets. Arguments H and r are not used in this context. See references [1] and [2] and the Examples for a more detailed discussion.

In the setting of a multivariate linear model, $X = A\Psi + U$, criteria Ccr12, Tau2, Xi2 and Zeta2 can be used to select subsets according to their contribution to an effect characterized by the violation of a reference hypothesis, $C\Psi = 0$ (see reference [3] for further details). In this setting, arguments mat and H should be set respectively to the usual Total (Hypothesis + Error) and Hypothesis, Sum of Squares and Cross-Products (SSCP) matrices. Argument r should be set to the expected rank of H. Currently, for reasons of computational efficiency, criterion Ccr12 is available only when r $\leq$ 3. Particular cases in this setting include Linear Discriminant Analyis (LDA), Linear Regression Analysis (LRA), Canonical Correlation Analysis (CCA) with one set of variables fixed and several extensions of these and other classical multivariate methodologies.

In the setting of a generalized linear model, criterion Wald can be used to select subsets according to the (lack of) significance of the discarded variables, as measured by the respective Wald's statistic (see reference [4] for further details). In this setting arguments mat and H should be set respectively to FI and FI %*% b %*% t(b) %*% FI, where b is a column vector of variable coefficient estimates and FI is an estimate of the corresponding Fisher information matrix.

The auxiliary functions lmHmat, ldaHmat glhHmat and glmHmat are provided to automatically create the matrices mat and H in all the cases considered.

Value

A list with five items:

subsets	An nsol x kmax x length(kmin:kmax) 3-dimensional array, giving for each cardinality (dimension 3) and each solution (dimension 1) the list of variables (referenced by their row/column numbers in matrix mat) in the subset (dimension 2). (For cardinalities smaller than kmax, the extra final positions are set to zero).
values	An nsol x length(kmin:kmax) matrix, giving for each cardinality (columns), the criterion values of the nsol (rows) subsets obtained.
bestvalues	A length(kmin:kmax) vector giving the best values of the criterion obtained for each cardinality. If improvement is TRUE, these values result from the final restricted local search algorithm (and may therefore exceed the largest value for that cardinality in values).
bestsets	A length(kmin:kmax) x kmax matrix, giving, for each cardinality (rows), the variables (referenced by their row/column numbers in matrix mat) in the best k-subset that was found.
call	The function call which generated the output.

References

[1] Cadima, J., Cerdeira, J. Orestes and Minhoto, M. (2004) Computational aspects of algorithms for variable selection in the context of principal components. *Computational Statistics and Data Analysis*, 47, 225-236.

[2] Cadima, J. and Jolliffe, I.T. (2001). Variable Selection and the Interpretation of Principal Subspaces, *Journal of Agricultural, Biological and Environmental Statistics*, Vol. 6, 62-79.

[3] Duarte Silva, A.P. (2001) Efficient Variable Screening for Multivariate Analysis, *Journal of Multivariate Analysis*, Vol. 76, 35-62.

[4] Lawless, J. and Singhal, K. (1978). Efficient Screening of Nonnormal Regression Models, *Biometrics*, Vol. 34, 318-327.

See Also

rm.coef, rv.coef, gcd.coef, tau2.coef, xi2.coef, zeta2.coef, ccr12.coef, genetic, anneal, eleaps, trim.matrix, lmHmat, ldaHmat, glhHmat, glmHmat.

Examples

```
## --------------------------------------------------------------------

##
## (1) For illustration of use, a small data set with very few iterations
## of the algorithm, using the RM criterion.
##

data(swiss)
anneal(cor(swiss),2,3,nsol=4,niter=10,criterion="RM")

##$subsets
##, , Card.2
##
##           Var.1 Var.2 Var.3
##Solution 1     3     6     0
##Solution 2     4     5     0
##Solution 3     1     2     0
##Solution 4     3     6     0
##
##, , Card.3
##
##           Var.1 Var.2 Var.3
##Solution 1     4     5     6
##Solution 2     3     5     6
##Solution 3     3     4     6
##Solution 4     4     5     6
##
##
##$values
##              card.2    card.3
##Solution 1 0.8016409 0.9043760
##Solution 2 0.7982296 0.8769672
##Solution 3 0.7945390 0.8777509
##Solution 4 0.8016409 0.9043760
##
##$bestvalues
##   Card.2    Card.3
##0.8016409 0.9043760
##
##$bestsets
```

```
##       Var.1 Var.2 Var.3
##Card.2     3     6     0
##Card.3     4     5     6
##
##$call
##anneal(cor(swiss), 2, 3, nsol = 4, niter = 10, criterion = "RM")

## --------------------------------------------------------------------

##
## (2) An example excluding variable number 6 from the subsets.
##

data(swiss)
anneal(cor(swiss),2,3,nsol=4,niter=10,criterion="RM",exclude=c(6))

##$subsets
##, , Card.2
##
##           Var.1 Var.2 Var.3
##Solution 1     4     5     0
##Solution 2     4     5     0
##Solution 3     4     5     0
##Solution 4     4     5     0
##
##, , Card.3
##
##           Var.1 Var.2 Var.3
##Solution 1     1     2     5
##Solution 2     1     2     5
##Solution 3     1     2     5
##Solution 4     1     4     5
##
##
##$values
##              card.2    card.3
##Solution 1 0.7982296 0.8791856
##Solution 2 0.7982296 0.8791856
##Solution 3 0.7982296 0.8791856
##Solution 4 0.7982296 0.8686515
##
##$bestvalues
##   Card.2    Card.3
##0.7982296 0.8791856
##
##$bestsets
##       Var.1 Var.2 Var.3
##Card.2     4     5     0
##Card.3     1     2     5
##
##$call
##anneal(cor(swiss), 2, 3, nsol = 4, niter = 10, criterion = "RM",
##     exclude=c(6))
```

```
## --------------------------------------------------------------------

## (3) An example specifying initial solutions: using the subsets produced
## by simulated annealing for one criterion (RM, by default) as initial
## solutions for the simulated annealing search with a different criterion.

data(swiss)
rmresults<-anneal(cor(swiss),2,3,nsol=4,niter=10, setseed=TRUE)
anneal(cor(swiss),2,3,nsol=4,niter=10,criterion="gcd",
initialsol=rmresults$subsets)

##$subsets
##, , Card.2
##
##            Var.1 Var.2 Var.3
##Solution 1     3     6     0
##Solution 2     3     6     0
##Solution 3     3     6     0
##Solution 4     3     6     0
##
##, , Card.3
##
##            Var.1 Var.2 Var.3
##Solution 1     4     5     6
##Solution 2     4     5     6
##Solution 3     3     4     6
##Solution 4     4     5     6
##
##
##$values
##               card.2   card.3
##Solution 1 0.8487026 0.925372
##Solution 2 0.8487026 0.925372
##Solution 3 0.8487026 0.798864
##Solution 4 0.8487026 0.925372
##
##$bestvalues
##   Card.2    Card.3
##0.8487026 0.9253720
##
##$bestsets
##        Var.1 Var.2 Var.3
##Card.2     3     6     0
##Card.3     4     5     6
##
##$call
##anneal(cor(swiss), 2, 3, nsol = 4, niter = 10, criterion = "gcd",
##    initialsol = rmresults$subsets)

## --------------------------------------------------------------------

## (4) An example of subset selection in the context of Multiple Linear
```

```
## Regression. Variable 5 (average car price) in the Cars93 MASS library
## data set is regressed on 13 other variables.  A best subset of linear
## predictors is sought, using the "TAU_2" criterion which, in the case
## of a Linear Regression, is merely  the standard Coefficient of Determination,
## R^2 (like the other three criteria for the multivariate linear hypothesis,
## "XI_2", "CCR1_2" and "ZETA_2").

library(MASS)
data(Cars93)
CarsHmat <- lmHmat(Cars93[,c(7:8,12:15,17:22,25)],Cars93[,5])

names(Cars93[,5,drop=FALSE])
##  [1] "Price"

colnames(CarsHmat$mat)

##  [1] "MPG.city"           "MPG.highway"        "EngineSize"
##  [4] "Horsepower"         "RPM"                "Rev.per.mile"
##  [7] "Fuel.tank.capacity" "Passengers"         "Length"
## [10] "Wheelbase"          "Width"              "Turn.circle"
## [13] "Weight"

anneal(CarsHmat$mat, kmin=4, kmax=6, H=CarsHmat$H, r=1, crit="tau2")

## $subsets
## , , Card.4
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     4     5    10    11     0     0
##
## , , Card.5
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     4     5    10    11    12     0
##
## , , Card.6
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     4     5     9    10    11    12
##
##
## $values
##               card.4    card.5   card.6
## Solution 1 0.7143794 0.7241457 0.731015
##
## $bestvalues
##    Card.4    Card.5    Card.6
## 0.7143794 0.7241457 0.7310150
##
## $bestsets
##        Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Card.4     4     5    10    11     0     0
## Card.5     4     5    10    11    12     0
```

```
## Card.6     4     5     9    10    11    12
##
## $call
## anneal(mat = CarsHmat$mat, kmin = 4, kmax = 6, criterion = "xi2",
##     H = CarsHmat$H, r = 1)
##

## -------------------------------------------------------------------

## (5) A Linear Discriminant Analysis example with a very small data set.
## We consider the Iris data and three groups, defined by species (setosa,
## versicolor and virginica). The goal is to select the 2- and 3-variable
## subsets that are optimal for the linear discrimination (as measured
## by the "CCR1_2" criterion).


data(iris)
irisHmat <- ldaHmat(iris[1:4],iris$Species)
anneal(irisHmat$mat,kmin=2,kmax=3,H=irisHmat$H,r=2,crit="ccr12")

## $subsets
## , , Card.2
##
##            Var.1 Var.2 Var.3
## Solution 1     1     3     0
##
## , , Card.3
##
##            Var.1 Var.2 Var.3
## Solution 1     2     3     4
##
##
## $values
##               card.2   card.3
## Solution 1 0.9589055 0.967897
##
## $bestvalues
##    Card.2    Card.3
## 0.9589055 0.9678971
##
## $bestsets
##        Var.1 Var.2 Var.3
## Card.2     1     3     0
## Card.3     2     3     4
##
## $call
## anneal(irisHmat$mat,kmin=2,kmax=3,H=irisHmat$H,r=2,crit="ccr12")
##

## -------------------------------------------------------------------

## (6) An example of subset selection in the context of a Canonical
## Correlation Analysis. Two groups of variables within the Cars93
```

```
## MASS library data set are compared. The goal is to select 4- to
## 6-variable subsets of the 13-variable 'X' group that are optimal in
## terms of preserving the canonical correlations, according to the
## "XI_2" criterion (Warning: the 3-variable 'Y' group is kept
## intact; subset selection is carried out in the 'X'
## group only). The 'tolsym' parameter is used to relax the symmetry
## requirements on the effect matrix H which, for numerical reasons,
## is slightly asymmetric. Since corresponding off-diagonal entries of
## matrix H are different, but by less than tolsym, H is replaced
## by its symmetric part: (H+t(H))/2.

library(MASS)
data(Cars93)
CarsHmat <- lmHmat(Cars93[,c(7:8,12:15,17:22,25)],Cars93[,4:6])

names(Cars93[,4:6])
## [1] "Min.Price" "Price"     "Max.Price"

colnames(CarsHmat$mat)

##  [1] "MPG.city"           "MPG.highway"        "EngineSize"
##  [4] "Horsepower"         "RPM"                "Rev.per.mile"
##  [7] "Fuel.tank.capacity" "Passengers"         "Length"
## [10] "Wheelbase"          "Width"              "Turn.circle"
## [13] "Weight"

anneal(CarsHmat$mat, kmin=4, kmax=6, H=CarsHmat$H, r=CarsHmat$r,
crit="tau2" , tolsym=1e-9)

## $subsets
## , , Card.4
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     4     9    10    11     0     0
##
## , , Card.5
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     3     4     9    10    11     0
##
## , , Card.6
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     3     4     5     9    10    11
##
##
## $values
##               card.4    card.5    card.6
## Solution 1 0.2818772 0.2943742 0.3057831
##
## $bestvalues
##    Card.4    Card.5    Card.6
## 0.2818772 0.2943742 0.3057831
```

```
##
## $bestsets
##        Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Card.4     4     9    10    11     0     0
## Card.5     3     4     9    10    11     0
## Card.6     3     4     5     9    10    11
##
## $call
## anneal(mat = CarsHmat$mat, kmin = 4, kmax = 6, criterion = "xi2",
##     H = CarsHmat$H, r = CarsHmat$r, tolsym = 1e-09)
##
## Warning message:
##
##  The effect description matrix (H) supplied was slightly asymmetric:
##  symmetric entries differed by up to 3.63797880709171e-12.
##  (less than the 'tolsym' parameter).
##  The H matrix has been replaced by its symmetric part.
##  in: validnovcrit(mat, criterion, H, r, p, tolval, tolsym)

## --------------------------------------------------------------------

##  (7) An example of variable selection in the context of a logistic
##  regression model. We consider the last 100 observations of
##  the iris data set (versicolor and verginica species) and try
##  to find the best variable subsets for the model that takes species
##  as response variable.

data(iris)
iris2sp <- iris[iris$Species != "setosa",]
logrfit <- glm(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
iris2sp,family=binomial)
Hmat <- glmHmat(logrfit)
anneal(Hmat$mat,1,3,H=Hmat$H,r=1,criterion="Wald")

## $subsets
## , , Card.1
##
##           Var.1 Var.2 Var.3
## Solution 1     4     0     0

## , , Card.2

##            Var.1 Var.2 Var.3
## Solution 1     1     3     0

## , , Card.3

##            Var.1 Var.2 Var.3
## Solution 1     2     3     4


## $values
##              card.1   card.2   card.3
```

```
## Solution 1 4.894554 3.522885 1.060121

## $bestvalues
##   Card.1   Card.2   Card.3
## 4.894554 3.522885 1.060121

## $bestsets
##        Var.1 Var.2 Var.3
## Card.1     4     0     0
## Card.2     1     3     0
## Card.3     2     3     4

## $call
## anneal(mat = Hmat$mat, kmin = 1, kmax = 3, criterion = "Wald",
##     H = Hmat$H, r = 1)
## --------------------------------------------------------------------

## It should be stressed that, unlike other criteria in the
## subselect package, the Wald criterion is not bounded above by
## 1 and is a decreasing function of subset quality, so that the
## 3-variable subsets do, in fact, perform better than their smaller-sized
## counterparts.
```

---

ccr12.coef — *First Squared Canonical Correlation for a multivariate linear hypothesis*

Description

Computes the first squared canonical correlation. The maximization of this criterion is equivalent to the maximization of the Roy first root.

Usage

```
ccr12.coef(mat, H, r, indices,
tolval=10*.Machine$double.eps, tolsym=1000*.Machine$double.eps)
```

Arguments

`mat`	the Variance or Total sums of squares and products matrix for the full data set.
`H`	the Effect description sums of squares and products matrix (defined in the same way as the `mat` matrix).
`r`	the Expected rank of the H matrix. See the `Details` section below.
`indices`	a numerical vector, matrix or 3-d array of integers giving the indices of the variables in the subset. If a matrix is specified, each row is taken to represent a different *k*-variable subset. If a 3-d array is given, it is assumed that the third dimension corresponds to different cardinalities.

`tolval` the tolerance level to be used in checks for ill-conditioning and positive-definiteness of the 'total' and 'effects' (H) matrices. Values smaller than `tolval` are considered equivalent to zero.

`tolsym` the tolerance level for symmetry of the covariance/correlation/total matrix and for the effects (`H`) matrix. If corresponding matrix entries differ by more than this value, the input matrices will be considered asymmetric and execution will be aborted. If corresponding entries are different, but by less than this value, the input matrix will be replaced by its symmetric part, i.e., input matrix A becomes (A+t(A))/2.

Details

Different kinds of statistical methodologies are considered within the framework, of a multivariate linear model:

$$X = A\Psi + U$$

where X is the (nxp) data matrix of original variables, A is a known (nxp) design matrix, Ψ an (qxp) matrix of unknown parameters and U an (nxp) matrix of residual vectors. The ccr_1^2 index is related to the traditional test statistic (the Roy first root) and measures the contribution of each subset to an Effect characterized by the violation of a linear hypothesis of the form $C\Psi = 0$, where C is a known cofficient matrix of rank r. The Roy first root is the first eigen value of HE^{-1}, where H is the Effect matrix and E is the Error matrix. The index ccr_1^2 is related to the Roy first root (λ_1) by:

$$ccr_1^2 = \frac{\lambda_1}{1 + \lambda_1}$$

The fact that `indices` can be a matrix or 3-d array allows for the computation of the ccr_1^2 values of subsets produced by the search functions `anneal`, `genetic`, `improve` and `anneal` (whose output option `$subsets` are matrices or 3-d arrays), using a different criterion (see the example below).

Value

The value of the ccr_1^2 coefficient.

Examples

```
## 1) A Linear Discriminant Analysis example with a very small data set.
## We considered the Iris data and three groups,
## defined by species (setosa, versicolor and virginica).

data(iris)
irisHmat <- ldaHmat(iris[1:4],iris$Species)
ccr12.coef(irisHmat$mat,H=irisHmat$H,r=2,c(1,3))
## [1]  0.9589055

## ---------------------------------------------------------------

## 2) An example computing the value of the ccr1_2 criteria for two
## subsets produced when the anneal function attempted to optimize
## the zeta_2 criterion (using an absurdly small number of iterations).
```

```
zetaresults<-anneal(irisHmat$mat,2,nsol=2,niter=2,criterion="zeta2",
H=irisHmat$H,r=2)
ccr12.coef(irisHmat$mat,H=irisHmat$H,r=2,zetaresults$subsets)

##              Card.2
##Solution 1 0.9526304
##Solution 2 0.9558787

## --------------------------------------------------------------
```

---

eleaps *A Leaps and Bounds Algorithm for finding the best variable subsets*

Description

An exact Algorithm for optimizing criteria that measure the quality of k-dimensional variable subsets as approximations to a given set of variables, or to a set of its Principal Components.

Usage

```
eleaps(mat,kmin=length(include)+1,kmax=ncol(mat)-length(exclude)-1,nsol=1,
exclude=NULL,include=NULL,criterion="default",pcindices="first_k",timelimit=15,
H=NULL,r=0, tolval=1000*.Machine$double.eps,
tolsym=1000*.Machine$double.eps,maxaperr=1E-4)
```

Arguments

`mat`	a covariance/correlation, information or sums of squares and products matrix of the variables from which the k-subset is to be selected. See the `Details` section below.
`kmin`	the cardinality of the smallest subset that is wanted.
`kmax`	the cardinality of the largest subset that is wanted.
`nsol`	the number of different subsets of each cardinality that are requested .
`exclude`	a vector of variables (referenced by their row/column numbers in matrix `mat`) that are to be forcibly excluded from the subsets.
`include`	a vector of variables (referenced by their row/column numbers in matrix `mat`) that are to be forcibly included in the subsets.
`criterion`	Character variable, which indicates which criterion is to be used in judging the quality of the subsets. Currently, the "Rm", "Rv", "Gcd", "Tau2", "Xi2", "Zeta2", "Ccr12" and "Wald" criteria are supported (see the `Details` section, the `References` and the links `rm.coef`, `rv.coef`, `gcd.coef`, `tau2.coef`, `xi2.coef`, `zeta2.coef`, `ccr12.coef` and `wald.coef` for further details). The default criterion is "Rm" if parameter `r` is zero (exploratory and PCA problems), "Wald" if `r` is equal to one and `mat` has a "FisherI" attribute set to TRUE (generalized linear models), and "Tau2" otherwise (multivariate linear model framework).

pcindices — either a vector of ranks of Principal Components that are to be used for comparison with the k-variable subsets (for the Gcd criterion only, see `gcd.coef`) or the default text `first_k`. The latter will associate PCs 1 to k with each cardinality k that has been requested by the user.

timelimit — a user specified limit (in seconds) for the maximum time allowed to conduct the search. After this limit is exceeded, `eleaps` exits with a waring message stating that it was not possible to find the otpimal subsets within the allocated time.

H — Effect description matrix. Not used with the Rm, Rv or Gcd criteria, hence the NULL default value. See the `Details` section below.

r — Expected rank of the effects (`H`) matrix. Not used with the Rm, Rv or Gcd criteria. See the `Details` section below.

tolval — the tolerance level for the reciprocal of the 2-norm condition number of the correlation/covariance or sums of squares matrix, i.e., for the ratio of the smallest to the largest eigenvalue of the input matrix. Matrices with a reciprocal of the condition number smaller than `tolval` will activate a restricted-search (for well conditioned sets as defined by the value of the `maxaperr` argument) algorithm.

tolsym — the tolerance level for symmetry of the covariance/correlation/total matrix and for the effects (`H`) matrix. If corresponding matrix entries differ by more than this value, the input matrices will be considered asymmetric and execution will be aborted. If corresponding entries are different, but by less than this value, the input matrix will be replaced by its symmetric part, i.e., input matrix A becomes (A+t(A))/2.

maxaperr — the tolerance level for the relative rounding error of the criterion. When a restricted search in employed subsets where a first order estimate of this error is higher than `maxaperr` will be excluded from the analysis.

Details

For each cardinality k (with k ranging from `kmin` to `kmax`), `eleaps` performs a branch and bound search for the best `nsol` variable subsets according to a specified criterion. Leaps implements Duarte Silva's adaptation (references [2] and [3]) of Furnival and Wilson's Leaps and Bounds Algorithm (reference [4]) for variable selection in Regression Analysis. If the search is not completed within a user defined time limit, `eleaps` exits with a warning message.

The user may force variables to be included and/or excluded from the *k*-subsets.

In order to improve computation times, the bulk of computations are carried out by C++ routines. Further details about the Algorithm can be found in references [2] and [3] and in the comments to the C++ code. A discussion of the criteria considered can be found in References [1] and [3].

The function checks for ill-conditioning of the input matrix (specifically, it checks whether the ratio of the input matrix's smallest and largest eigenvalues is less than `tolval`). For an ill-conditioned input matrix, the search is restricted to its well-conditioned subsets. The function `trim.matrix` may be used to obtain a well-conditioned input matrix.

In a general descriptive (Principal Components Analysis) setting, the three criteria Rm, Rv and Gcd can be used to select good k-variable subsets. Arguments `H` and `r` are not used in this context. See reference [1] and the `Examples` for a more detailed discussion.

In the setting of a multivariate linear model, $X = A\Psi + U$, criteria Ccr12, Tau2, Xi2 and Zeta2 can be used to select subsets according to their contribution to an effect characterized by the violation of a reference hypothesis, $C\Psi = 0$ (see reference [3] for further details). In this setting, arguments `mat` and `H` should be set respectively to the usual Total (Hypothesis + Error) and Hypothesis, Sum of Squares and Cross-Products (SSCP) matrices. Argument `r` should be set to the expected rank of `H`. Currently, for reasons of computational efficiency, criterion Ccr12 is available only when `r` $\leq$ 3. Particular cases in this setting include Linear Discriminant Analyis (LDA), Linear Regression Analysis (LRA), Canonical Correlation Analysis (CCA) with one set of variables fixed, and several extensions of these and other classical multivariate methodologies.

In the setting of a generalized linear model, criterion Wald can be used to select subsets according to the (lack of) significance of the discarded variables, as measured by the respective Wald's statistic (see reference [5] for further details). In this setting arguments `mat` and `H` should be set respectively to FI and `FI %*% b %*% t(b) %*% FI`, where `b` is a column vector of variable coefficient estimates and FI is an estimate of the corresponding Fisher information matrix.

The auxiliary functions `lmHmat`, `ldaHmat` `glhHmat` and `glmHmat` are provided to automatically create the matrices `mat` and `H` in all the cases considered.

Value

A list with five items:

`subsets` An `nsol` x `kmax` x length(`kmin`:`kmax`) 3-dimensional array, giving for each cardinality (dimension 3) and each solution (dimension 1) the list of variables (referenced by their row/column numbers in matrix `mat`) in the subset (dimension 2). (For cardinalities smaller than `kmax`, the extra final positions are set to zero).

`values` An `nsol` x length(`kmin`:`kmax`) matrix, giving for each cardinality (columns), the criterion values of the best `nsol` (rows) subsets according to the chosen criterion.

`bestvalues` A length(`kmin`:`kmax`) vector giving the overall best values of the criterion for each cardinality.

`bestsets` A length(`kmin`:`kmax`) x `kmax` matrix, giving, for each cardinality (rows), the variables (referenced by their row/column numbers in matrix `mat`) in the best k-subset.

`call` The function call which generated the output.

References

[1] Cadima, J. and Jolliffe, I.T. (2001). Variable Selection and the Interpretation of Principal Subspaces, *Journal of Agricultural, Biological and Environmental Statistics*, Vol. 6, 62-79.

[2] Duarte Silva, A.P. (2001) Efficient Variable Screening for Multivariate Analysis, Journal of Multivariate Analysis Vol. 76, 35-62.

[3] Duarte Silva, A.P. (2002) Discarding Variables in a Principal Component Analysis: Algorithms for All-Subsets Comparisons, *Computational Statistics*, Vol. 17, 251-271.

[4] Furnival, G.M. and Wilson, R.W. (1974). Regressions by Leaps and Bounds, *Technometrics*, Vol. 16, 499-511.

[5] Lawless, J. and Singhal, K. (1978). Efficient Screening of Nonnormal Regression Models, *Biometrics*, Vol. 34, 318-327.

See Also

rm.coef, rv.coef, gcd.coef, tau2.coef, wald.coef, xi2.coef, zeta2.coef, ccr12.coef, anneal, genetic, anneal, trim.matrix, lmHmat, ldaHmat, glhHmat, glmHmat.

Examples

```
## --------------------------------------------------------------------

##
## 1) For illustration of use, a small data set.
## Subsets of variables of all cardinalities are sought using the
## RM criterion.
##

data(swiss)
eleaps(cor(swiss),nsol=3, criterion="RM")

##$subsets
##, , Card.1
##
##           Var.1 Var.2 Var.3 Var.4 Var.5
##Solution 1     3     0     0     0     0
##Solution 2     1     0     0     0     0
##Solution 3     4     0     0     0     0
##
##, , Card.2
##
##           Var.1 Var.2 Var.3 Var.4 Var.5
##Solution 1     3     6     0     0     0
##Solution 2     4     5     0     0     0
##Solution 3     1     2     0     0     0
##
##, , Card.3
##
##           Var.1 Var.2 Var.3 Var.4 Var.5
##Solution 1     4     5     6     0     0
##Solution 2     1     2     5     0     0
##Solution 3     3     4     6     0     0
##
##, , Card.4
##
##           Var.1 Var.2 Var.3 Var.4 Var.5
##Solution 1     2     4     5     6     0
##Solution 2     1     2     5     6     0
##Solution 3     1     4     5     6     0
##
##, , Card.5
##
##           Var.1 Var.2 Var.3 Var.4 Var.5
##Solution 1     1     2     3     5     6
##Solution 2     1     2     4     5     6
##Solution 3     2     3     4     5     6
```

```
##
##
##$values
##              card.1    card.2    card.3    card.4    card.5
##Solution 1 0.6729689 0.8016409 0.9043760 0.9510757 0.9804629
##Solution 2 0.6286185 0.7982296 0.8791856 0.9506434 0.9776338
##Solution 3 0.6286130 0.7945390 0.8777509 0.9395708 0.9752551
##
##$bestvalues
##   Card.1    Card.2    Card.3    Card.4    Card.5
##0.6729689 0.8016409 0.9043760 0.9510757 0.9804629
##
##$bestsets
##       Var.1 Var.2 Var.3 Var.4 Var.5
##Card.1     3     0     0     0     0
##Card.2     3     6     0     0     0
##Card.3     4     5     6     0     0
##Card.4     2     4     5     6     0
##Card.5     1     2     3     5     6
##
##$call
##eleaps(cor(swiss), nsol = 3, criterion="RM")


## --------------------------------------------------------------------

##
## 2) Asking only for 2- and 3- dimensional subsets and excluding
## variable number 6.
##

data(swiss)
eleaps(cor(swiss),2,3,exclude=6,nsol=3,criterion="rm")

##$subsets
##, , Card.2
##
##           Var.1 Var.2 Var.3
##Solution 1     4     5     0
##Solution 2     1     2     0
##Solution 3     1     3     0
##
##, , Card.3
##
##           Var.1 Var.2 Var.3
##Solution 1     1     2     5
##Solution 2     1     4     5
##Solution 3     2     4     5
##
##
##$values
##              card.2    card.3
##Solution 1 0.7982296 0.8791856
```

```
##Solution 2 0.7945390 0.8686515
##Solution 3 0.7755232 0.8628693
##
##$bestvalues
##   Card.2    Card.3
##0.7982296 0.8791856
##
##$bestsets
##       Var.1 Var.2 Var.3
##Card.2     4     5     0
##Card.3     1     2     5
##
##$call
##eleaps(cor(swiss), 2, 3, exclude = 6, nsol = 3, criterion = "gcd")

## -------------------------------------------------------------------

##
## 3) Searching for 2- and 3- dimensional subsets that best approximate
## the spaces generated by the first three Principal Components
##


data(swiss)
eleaps(cor(swiss),2,3,criterion="gcd",pcindices=1:3,nsol=3)

##$subsets
##, , Card.2
##
##           Var.1 Var.2 Var.3
##Solution 1     4     5     0
##Solution 2     5     6     0
##Solution 3     4     6     0
##
##, , Card.3
##
##           Var.1 Var.2 Var.3
##Solution 1     4     5     6
##Solution 2     3     5     6
##Solution 3     2     5     6
##
##
##$values
##              card.2    card.3
##Solution 1 0.7831827 0.9253684
##Solution 2 0.7475630 0.8459302
##Solution 3 0.7383665 0.8243032
##
##$bestvalues
##   Card.2    Card.3
##0.7831827 0.9253684
##
##$bestsets
```

```
##       Var.1 Var.2 Var.3
##Card.2     4     5     0
##Card.3     4     5     6
##
##$call
##eleaps(cor(swiss), 2, 3, criterion = "gcd", pcindices = 1:3, nsol = 3)


## --------------------------------------------------------------------

##
## 4) An example of subset selection in the context of Multiple Linear
## Regression. Variable 5 (average car price) in the Cars93 MASS library
## data set is regressed on 13 other variables. A best subset of linear
## predictors is sought, using the default criterion ("TAU_2") which,
## in the case of a Linear Regression, is merely  the standard Coefficient
## of Determination, R^2 (as are the other three criteria for the
## multivariate linear hypothesis, "XI_2", "CCR1_2" and "ZETA_2").
##

library(MASS)
data(Cars93)
CarsHmat <- lmHmat(Cars93[,c(7:8,12:15,17:22,25)],Cars93[,5])

names(Cars93[,5,drop=FALSE])
## [1] "Price"

colnames(CarsHmat$mat)

##  [1] "MPG.city"           "MPG.highway"        "EngineSize"
##  [4] "Horsepower"         "RPM"                "Rev.per.mile"
##  [7] "Fuel.tank.capacity" "Passengers"         "Length"
## [10] "Wheelbase"          "Width"              "Turn.circle"
## [13] "Weight"


eleaps(CarsHmat$mat, kmin=4, kmax=6, H=CarsHmat$H, r=1)

## $subsets
## , , Card.4
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     4     5    10    11     0     0
##
## , , Card.5
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     4     5    10    11    12     0
##
## , , Card.6
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     4     5     9    10    11    12
```

```
##
##
## $values
##              card.4    card.5   card.6
## Solution 1 0.7143794 0.7241457 0.731015
##
## $bestvalues
##    Card.4    Card.5    Card.6
## 0.7143794 0.7241457 0.7310150
##
## $bestsets
##        Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Card.4     4     5    10    11     0     0
## Card.5     4     5    10    11    12     0
## Card.6     4     5     9    10    11    12
##

## ----------------------------------------------------------------------

## 5) A Linear Discriminant Analysis example with a very small data set.
## We consider the Iris data and three groups, defined by species (setosa,
## versicolor and virginica). The goal is to select the 2- and 3-variable
## subsets that are optimal for the linear discrimination (as measured
## by the "CCR1_2" criterion).

data(iris)
irisHmat <- ldaHmat(iris[1:4],iris$Species)
eleaps(irisHmat$mat,kmin=2,kmax=3,H=irisHmat$H,r=2,crit="ccr12")

## $subsets
## , , Card.2
##
##            Var.1 Var.2 Var.3
## Solution 1     1     3     0
##
## , , Card.3
##
##            Var.1 Var.2 Var.3
## Solution 1     2     3     4
##
##
## $values
##              card.2   card.3
## Solution 1 0.9589055 0.967897
##
## $bestvalues
##    Card.2    Card.3
## 0.9589055 0.9678971
##
## $bestsets
##        Var.1 Var.2 Var.3
## Card.2     1     3     0
## Card.3     2     3     4
```

```
## ----------------------------------------------------------------------

## 6) An example of subset selection in the context of a Canonical
## Correlation Analysis. Two groups of variables within the Cars93
## MASS library data set are compared. The goal is to select 4- to
## 6-variable subsets of the 13-variable 'X' group that are optimal in
## terms of preserving the canonical correlations, according to the
## "ZETA_2" criterion (Warning: the 3-variable 'Y' group is kept
## intact; subset selection is carried out in the 'X'
## group only).  The 'tolsym' parameter is used to relax the symmetry
## requirements on the effect matrix H which, for numerical reasons,
## is slightly asymmetric. Since corresponding off-diagonal entries of
## matrix H are different, but by less than tolsym, H is replaced
## by its symmetric part: (H+t(H))/2.

library(MASS)
data(Cars93)
CarsHmat <- lmHmat(Cars93[,c(7:8,12:15,17:22,25)],Cars93[,4:6])

names(Cars93[,4:6])
## [1] "Min.Price" "Price"     "Max.Price"

##  colnames(CarsHmat$mat)


##  [1] "MPG.city"           "MPG.highway"        "EngineSize"
##  [4] "Horsepower"         "RPM"                "Rev.per.mile"
##  [7] "Fuel.tank.capacity" "Passengers"         "Length"
## [10] "Wheelbase"          "Width"              "Turn.circle"
## [13] "Weight"


eleaps(CarsHmat$mat, kmin=4, kmax=6, H=CarsHmat$H, r=3,
crit="zeta2", tolsym=1e-9)

## $subsets
## , , Card.4
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     3     4    10    11     0     0
##
## , , Card.5
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     4     5     9    10    11     0
##
## , , Card.6
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     4     5     9    10    11    12
##
```

```
##
## $values
##               card.4    card.5    card.6
## Solution 1 0.4827353 0.5018922 0.5168627
##
## $bestvalues
##    Card.4    Card.5    Card.6
## 0.4827353 0.5018922 0.5168627
##
## $bestsets
##        Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Card.4     3     4    10    11     0     0
## Card.5     4     5     9    10    11     0
## Card.6     4     5     9    10    11    12
##
## Warning message:
##
##  The effect description matrix (H) supplied was slightly asymmetric:
##  symmetric entries differed by up to 3.63797880709171e-12.
##  (less than the 'tolsym' parameter).
##  The H matrix has been replaced by its symmetric part.
##  in: validnovcrit(mat, criterion, H, r, p, tolval, tolsym)

## --------------------------------------------------------------------


##  7) An example of variable selection in the context of a logistic
##  regression model. We consider the last 100 observations of
##  the iris data set (versicolor an verginica species) and try
##  to find the best variable subsets for the model that takes species
##  as response variable.

data(iris)
iris2sp <- iris[iris$Species != "setosa",]
logrfit <- glm(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
iris2sp,family=binomial)
Hmat <- glmHmat(logrfit)
eleaps(Hmat$mat,H=Hmat$H,r=1,criterion="Wald",nsol=3)

## $subsets
## , , Card.1

##            Var.1 Var.2 Var.3
## Solution 1     4     0     0
## Solution 2     1     0     0
## Solution 3     3     0     0

## , , Card.2

##           Var.1 Var.2 Var.3
## Solution 1     1     3     0
## Solution 2     3     4     0
## Solution 3     2     4     0
```

```
## , , Card.3

##            Var.1 Var.2 Var.3
## Solution 1     2     3     4
## Solution 2     1     3     4
## Solution 3     1     2     3


## $values
##              card.1   card.2   card.3
## Solution 1 4.894554 3.522885 1.060121
## Solution 2 5.147360 3.952538 2.224335
## Solution 3 5.161553 3.972410 3.522879

## $bestvalues
##   Card.1   Card.2   Card.3
## 4.894554 3.522885 1.060121

## $bestsets
##        Var.1 Var.2 Var.3
## Card.1     4     0     0
## Card.2     1     3     0
## Card.3     2     3     4

## $call
## eleaps(mat = Hmat$mat, nsol = 3, criterion = "Wald", H = Hmat$H,
##    r = 1)
## --------------------------------------------------------------------

## It should be stressed that, unlike other criteria in the
## subselect package, the Wald criterion is not bounded above by
## 1 and is a decreasing function of subset quality, so that the
## 3-variable subsets do, in fact, perform better than their smaller-sized
## counterparts.
```

`farm` *Sixty-two economic indicators from 99 Portuguese farms.*

Description

This data set is a very small subset of economic data regarding Portuguese farms in the mid-1990s, from Portugal's Ministry of Agriculture

Usage

```
farm
```

Format

A 99x62 matrix. The 62 columns are numeric economic indicators, referenced by their database code. Monetary units are in thousands of Escudos (Portugal's pre-Euro currency).

Column Number	Column Name	Units	Description
[,1]	R15	1000 Escudos	Total Standard Gross Margins (SGM)
[,2]	R24	Hectares	Total land surface
[,3]	R35	Hectares	Total cultivated surface
[,4]	R36	Man Work Units	Total Man Work Units
[,5]	R46	1000 Escudos	Land Capital
[,6]	R59	1000 Escudos	Total Capital (without forests)
[,7]	R65	1000 Escudos	Total Loans and Debts
[,8]	R72	1000 Escudos	Total Investment
[,9]	R79	1000 Escudos	Subsidies for Investment
[,10]	R86	1000 Escudos	Gross Plant Product Formation
[,11]	R91	1000 Escudos	Gross Animal Product Formation
[,12]	R104	1000 Escudos	Current Subsidies
[,13]	R110	1000 Escudos	Wheat Production
[,14]	R111	1000 Escudos	Maize Production
[,15]	R113	1000 Escudos	Other Cereals (except rice) Production
[,16]	R114	1000 Escudos	Dried Legumes Production
[,17]	R115	1000 Escudos	Potato Production
[,18]	R116	1000 Escudos	Industrial horticulture and Melon Production
[,19]	R117	1000 Escudos	Open-air horticultural Production
[,20]	R118	1000 Escudos	Horticultural forcing Production
[,21]	R119	1000 Escudos	Flower Production
[,22]	R121	1000 Escudos	Sub-products Production
[,23]	R122	1000 Escudos	Fruit Production
[,24]	R123	1000 Escudos	Olive Production
[,25]	R124	1000 Escudos	Wine Production
[,26]	R125	1000 Escudos	Horses
[,27]	R126	1000 Escudos	Bovines (excluding milk)
[,28]	R127	1000 Escudos	Milk and dairy products
[,29]	R129	1000 Escudos	Sheep
[,30]	R132	1000 Escudos	Goats
[,31]	R135	1000 Escudos	Pigs
[,32]	R137	1000 Escudos	Birds
[,33]	R140	1000 Escudos	Bees
[,34]	R142	1000 Escudos	Other animals (except rabbits)
[,35]	R144	1000 Escudos	Wood production
[,36]	R145	1000 Escudos	Other forest products (except cork)
[,37]	R146	Hectares	Land surface affected to cereals
[,38]	R151	Hectares	Land surface affected to dry legumes
[,39]	R152	Hectares	Land surface affected to potatos
[,40]	R158	Hectares	Land surface affected to fruits
[,41]	R159	Hectares	Land surface affected to olive trees
[,42]	R160	Hectares	Land surface affected to vineyards
[,43]	R164	Hectares	Fallow land surface area

[,44]	R166	Hectares	Forest surface area
[,45]	R168	Head	Bovines
[,46]	R174	Head	Adult sheep
[,47]	R176	Head	Adult goats
[,48]	R178	Head	Adult pigs
[,49]	R209	Kg/hectare	Maize yield
[,50]	R211	Kg/hectare	Barley yield
[,51]	R214	Kg/hectare	Potato yield
[,52]	R215	L/cow/year	Cow milk productivity
[,53]	R233	1000 Escudos	Wages and social expenditure
[,54]	R237	1000 Escudos	Taxes and tariffs
[,55]	R245	1000 Escudos	Interest and financial costs
[,56]	R250	1000 Escudos	Total real costs
[,57]	R252	1000 Escudos	Gross Product
[,58]	R256	1000 Escudos	Gross Agricultural Product
[,59]	R258	1000 Escudos	Gross Value Added (GVA)
[,60]	R263	1000 Escudos	Final Results
[,61]	R270	1000 Escudos	Family labour income
[,62]	R271	1000 Escudos	Capital Income

Source

Obtained directly from the source.

`gcd.coef` *Computes Yanai's GCD in the context of the variable-subset selection problem*

Description

Computes Yanai's Generalized Coefficient of Determination for the similarity of the subspaces spanned by a subset of variables and a subset of the full data set's Principal Components.

Usage

```
gcd.coef(mat, indices, pcindices =  NULL)
```

Arguments

`mat` the full data set's covariance (or correlation) matrix.

`indices` a numerical vector, matrix or 3-d array of integers giving the indices of the variables in the subset. If a matrix is specified, each row is taken to represent a different *k*-variable subset. If a 3-d array is given, it is assumed that the third dimension corresponds to different cardinalities.

`pcindices` a numerical vector of indices of Principal Components. By default, the first *k* PCs are chosen, where *k* is the cardinality of the subset of variables whose criterion value is being computed. **If a vector of PCs is specified by the user, those PCs will be used for all cardinalities that were requested**.

Details

Computes Yanai's Generalized Coefficient of Determination for the similarity of the subspaces spanned by a subset of variables (specified by `indices`) and a subset of the full-data set's Principal Components (specified by `pcindices`). Input data is expected in the form of a (co)variance or correlation matrix. If a non-square matrix is given, it is assumed to be a data matrix, and its correlation matrix is used as input. The number of variables (k) and of PCs (q) does not have to be the same.

Yanai's GCD is defined as:

$$GCD = \frac{\mathrm{tr}(P_v \cdot P_c)}{\sqrt{k \cdot q}}$$

where P_v and P_c are the matrices of orthogonal projections on the subspaces spanned by the k-variable subset and by the q-Principal Component subset, respectively.

This definition is equivalent to:

$$GCD = \frac{1}{\sqrt{kq}} \sum_i (r_m)_i^2$$

where $(r_m)_i$ stands for the multiple correlation between the `i`-th Principal Component and the k-variable subset, and the sum is carried out over the q PCs (i=1,...,q) selected.

These definitions are also equivalent to the expression used in the code, which only requires the covariance (or correlation) matrix of the data under consideration.

The fact that `indices` can be a matrix or 3-d array allows for the computation of the GCD values of subsets produced by the search functions `anneal`, `genetic` and `improve` (whose output option `$subsets` are matrices or 3-d arrays), using a different criterion (see the example below).

Value

The value of the GCD coefficient.

References

Cadima, J. and Jolliffe, I.T. (2001), "Variable Selection and the Interpretation of Principal Subspaces", *Journal of Agricultural, Biological and Environmental Statistics*, Vol. 6, 62-79.

Ramsay, J.O., ten Berge, J. and Styan, G.P.H. (1984), "Matrix Correlation", *Psychometrika*, 49, 403-423.

Examples

```
## An example with a very small data set.

data(iris3)
x<-iris3[,,1]
gcd.coef(cor(x),c(1,3))
## [1] 0.7666286
gcd.coef(cor(x),c(1,3),pcindices=c(1,3))
## [1] 0.584452
gcd.coef(cor(x),c(1,3),pcindices=1)
## [1] 0.6035127

## An example computing the GCDs of three subsets produced when the
```

```
## anneal function attempted to optimize the RV criterion (using an
## absurdly small number of iterations).

data(swiss)
rvresults<-anneal(cor(swiss),2,nsol=4,niter=5,criterion="Rv")
gcd.coef(cor(swiss),rvresults$subsets)

##             Card.2
##Solution 1 0.4962297
##Solution 2 0.7092591
##Solution 3 0.4748525
##Solution 4 0.4649259
```

---

genetic *Genetic Algorithm searching for an optimal k-variable subset*

Description

Given a set of variables, a Genetic Algorithm algorithm seeks a k-variable subset which is optimal, as a surrogate for the whole set, with respect to a given criterion.

Usage

```
genetic( mat, kmin, kmax = kmin, popsize = max(100,2*ncol(mat)), nger = 100,
mutate = FALSE, mutprob = 0.01, maxclone = 5, exclude = NULL,
include = NULL, improvement = TRUE, setseed= FALSE, criterion = "default",
pcindices = "first_k", initialpop = NULL, force = FALSE, H=NULL, r=0,
tolval=1000*.Machine$double.eps,tolsym=1000*.Machine$double.eps)
```

Arguments

`mat`	a covariance/correlation, information or sums of squares and products matrix of the variables from which the k-subset is to be selected. See the `Details` section below.
`kmin`	the cardinality of the smallest subset that is wanted.
`kmax`	the cardinality of the largest subset that is wanted.
`popsize`	integer variable indicating the size of the population.
`nger`	integer variable giving the number of generations for which the genetic algorithm will run.
`mutate`	logical variable indicating whether each child undergoes a mutation, with probability `mutprob`. By default, FALSE.
`mutprob`	variable giving the probability of each child undergoing a mutation, if `mutate` is TRUE. By default, 0.01. High values slow down the algorithm considerably and tend to replicate the same solution.

maxclone
: integer variable specifying the maximum number of identical replicates (clones) of individuals that is acceptable in the population. Serves to ensure that the population has sufficient genetic diversity, which is necessary to enable the algorithm to complete the specified number of generations. However, even maxclone=0 does not guarantee that there are no repetitions: only the offspring of couples are tested for clones. If any such clones are rejected, they are replaced by a k-variable subset chosen at random, without any further clone tests.

exclude
: a vector of variables (referenced by their row/column numbers in matrix `mat`) that are to be forcibly excluded from the subsets.

include
: a vector of variables (referenced by their row/column numbers in matrix `mat`) that are to be forcibly included in the subsets.

improvement
: a logical variable indicating whether or not the best final subset (for each cardinality) is to be passed as input to a local improvement algorithm (see function `improve`).

setseed
: logical variable indicating whether to fix an initial seed for the random number generator, which will be re-used in future calls to this function whenever setseed is again set to TRUE.

criterion
: Character variable, which indicates which criterion is to be used in judging the quality of the subsets. Currently, the "Rm", "Rv", "Gcd", "Tau2", "Xi2", "Zeta2", "ccr12" and "Wald" criteria are supported (see the `Details` section, the `References` and the links `rm.coef`, `rv.coef`, `gcd.coef`, `tau2.coef`, `xi2.coef`, `zeta2.coef` and `ccr12.coef` for further details). The default criterion is "Rm" if parameter `r` is zero (exploratory and PCA problems), "Wald" if `r` is equal to one and `mat` has a "FisherI" attribute set to TRUE (generalized linear models), and "Tau2" otherwise (multivariate linear model framework).

pcindices
: either a vector of ranks of Principal Components that are to be used for comparison with the k-variable subsets (for the Gcd criterion only, see `gcd.coef`) or the default text `first_k`. The latter will associate PCs 1 to *k* with each cardinality *k* that has been requested by the user.

initialpop
: vector, matrix or 3-d array of initial population for the genetic algorithm. If a *single cardinality* is required, `initialpop` may be a `popsize` x *k* matrix or a `popsize` x *k* x 1 array (as produced by the `$subsets` output value of any of the algorithm functions `anneal`, `genetic`, or `improve`). If *more than one cardinality* is requested, `initialpop` must be a `popsize` x `kmax` x `length(kmin:kmax)` 3-d array (as produced by the `$subsets` output value).

  If the `exclude` and/or `include` options are used, `initialpop` must also respect those requirements.

force
: a logical variable indicating whether, for large data sets (currently p > 400) the algorithm should proceed anyways, regardless of possible memory problems which may crash the R session.

H
: Effect description matrix. Not used with the Rm, Rv or Gcd criteria, hence the NULL default value. See the `Details` section below.

r
: Expected rank of the effects (`H`) matrix. Not used with the Rm, Rv or Gcd criteria. See the `Details` section below.

tolval
: the tolerance level for the reciprocal of the 2-norm condition number of the correlation/covariance matrix, i.e., for the ratio of the smallest to the largest eigenvalue of the input matrix. Matrices with a reciprocal of the condition number smaller than `tolval` will activate a restricted-search for well conditioned subsets.

tolsym
: the tolerance level for symmetry of the covariance/correlation/total matrix and for the effects (`H`) matrix. If corresponding matrix entries differ by more than this value, the input matrices will be considered asymmetric and execution will be aborted. If corresponding entries are different, but by less than this value, the input matrix will be replaced by its symmetric part, i.e., input matrix A becomes (A+t(A))/2.

Details

For each cardinality k (with k ranging from `kmin` to `kmax`), an initial population of `popsize` k-variable subsets is randomly selected from a full set of p variables. In each iteration, `popsize`/2 couples are formed from among the population and each couple generates a child (a new k-variable subset) which inherits properties of its parents (specifically, it inherits all variables common to both parents and a random selection of variables in the symmetric difference of its parents' genetic makeup). Each offspring may optionally undergo a mutation (in the form of a local improvement algorithm – see function `improve`), with a user-specified probability. The parents and offspring are ranked according to their criterion value, and the best `popsize` of these k-subsets will make up the next generation, which is used as the current population in the subsequent iteration.

The stopping rule for the algorithm is the number of generations (`nger`).

Optionally, the best *k*-variable subset produced by the Genetic Algorithm may be passed as input to a restricted local improvement algorithm, for possible further improvement (see function `improve`).

The user may force variables to be included and/or excluded from the *k*-subsets, and may specify an initial population.

For each cardinality *k*, the total number of calls to the procedure which computes the criterion values is $popsize + nger$ x $popsize/2$. These calls are the dominant computational effort in each iteration of the algorithm.

In order to improve computation times, the bulk of computations are carried out by a Fortran routine. Further details about the Genetic Algorithm can be found in Reference 1 and in the comments to the Fortran code (in the `src` subdirectory for this package). For datasets with a very large number of variables (currently p > 400), it is necessary to set the `force` argument to TRUE for the function to run, but this may cause a session crash if there is not enough memory available.

The function checks for ill-conditioning of the input matrix (specifically, it checks whether the ratio of the input matrix's smallest and largest eigenvalues is less than `tolval`). For an ill-conditioned input matrix, the search is restricted to its well-conditioned subsets. The function `trim.matrix` may be used to obtain a well-conditioned input matrix.

In a general descriptive (Principal Components Analysis) setting, the three criteria Rm, Rv and Gcd can be used to select good k-variable subsets. Arguments `H` and `r` are not used in this context. See references [1] and [2] and the `Examples` for a more detailed discussion.

In the setting of a multivariate linear model, $X = A\Psi + U$, criteria Ccr12, Tau2, Xi2 and Zeta2 can be used to select subsets according to their contribution to an effect characterized by the violation of a reference hypothesis, $C\Psi = 0$ (see reference [3] for further details). In this setting, arguments mat and H should be set respectively to the usual Total (Hypothesis + Error) and Hypothesis, Sum of Squares and Cross-Products (SSCP) matrices. Argument r should be set to the expected rank of H. Currently, for reasons of computational efficiency, criterion Ccr12 is available only when r $\leq$ 3. Particular cases in this setting include Linear Discriminant Analyis (LDA), Linear Regression Analysis (LRA), Canonical Correlation Analysis (CCA) with one set of variables fixed and several extensions of these and other classical multivariate methodologies.

In the setting of a generalized linear model, criterion Wald can be used to select subsets according to the (lack of) significance of the discarded variables, as measured by the respective Wald's statistic (see reference [4] for further details). In this setting arguments mat and H should be set respectively to FI and FI %*% b %*% t(b) %*% FI, where b is a column vector of variable coefficient estimates and FI is an estimate of the corresponding Fisher information matrix.

The auxiliary functions lmHmat, ldaHmat glhHmat and glmHmat are provided to automatically create the matrices mat and H in all the cases considered.

Value

A list with five items:

subsets
: A popsize x kmax x length(kmin:kmax) 3-dimensional array, giving for each cardinality (dimension 3) and each subset in the final population (dimension 1) the list of variables (referenced by their row/column numbers in matrix mat) in the subset (dimension 2). (For cardinalities smaller than kmax, the extra final positions are set to zero).

values
: A popsize x length(kmin:kmax) matrix, giving for each cardinality (columns), the (ordered) criterion values of the popsize (rows) subsets in the final generation.

bestvalues
: A length(kmin:kmax) vector giving the best values of the criterion obtained for each cardinality. If improvement is TRUE, these values result from the final restricted local search algorithm (and may therefore exceed the largest value for that cardinality in values).

bestsets
: A length(kmin:kmax) x kmax matrix, giving, for each cardinality (rows), the variables (referenced by their row/column numbers in matrix mat) in the best k-subset that was found.

call
: The function call which generated the output.

References

[1] Cadima, J., Cerdeira, J. Orestes and Minhoto, M. (2004) Computational aspects of algorithms for variable selection in the context of principal components. *Computational Statistics and Data Analysis*, 47, 225-236.

[2] Cadima, J. and Jolliffe, I.T. (2001). Variable Selection and the Interpretation of Principal Subspaces, *Journal of Agricultural, Biological and Environmental Statistics*, Vol. 6, 62-79.

[3] Duarte Silva, A.P. (2001) Efficient Variable Screening for Multivariate Analysis, *Journal of Multivariate Analysis*, Vol. 76, 35-62.

[4] Lawless, J. and Singhal, K. (1978). Efficient Screening of Nonnormal Regression Models, *Biometrics*, Vol. 34, 318-327.

See Also

rm.coef, rv.coef, gcd.coef, tau2.coef, xi2.coef, zeta2.coef, ccr12.coef, genetic, anneal, eleaps, trim.matrix, lmHmat, ldaHmat, glhHmat, glmHmat.

Examples

```
## --------------------------------------------------------------------

##
## 1) For illustration of use, a small data set with very few iterations
## of the algorithm.  Escoufier's 'RV' criterion is used to select variable
## subsets of size 3 and 4.
##

data(swiss)
genetic(cor(swiss),3,4,popsize=10,nger=5,criterion="Rv")

## For cardinality k=
##[1] 4
## there is not enough genetic diversity in generation number
##[1] 3
## for acceptable levels of consanguinity (couples differing by at least 2 genes).
## Try reducing the maximum acceptable number  of clones (maxclone) or
## increasing the population size (popsize)
## Best criterion value found so far:
##[1] 0.9557145
##$subsets
##, , Card.3
##
##            Var.1 Var.2 Var.3 Var.4
##Solution 1      1     2     3     0
##Solution 2      1     2     3     0
##Solution 3      1     2     3     0
##Solution 4      3     4     6     0
##Solution 5      3     4     6     0
##Solution 6      3     4     5     0
##Solution 7      3     4     5     0
##Solution 8      1     3     6     0
##Solution 9      1     3     6     0
##Solution 10     1     3     6     0
##
##, , Card.4
##
##            Var.1 Var.2 Var.3 Var.4
##Solution 1      2     4     5     6
##Solution 2      1     2     5     6
##Solution 3      1     2     3     5
##Solution 4      1     2     4     5
##Solution 5      1     2     4     5
##Solution 6      1     4     5     6
##Solution 7      1     4     5     6
##Solution 8      1     4     5     6
```

```
##Solution 9      1     3     4     5
##Solution 10     1     3     4     5
##
##
##$values
##               card.3    card.4
##Solution 1  0.9141995 0.9557145
##Solution 2  0.9141995 0.9485699
##Solution 3  0.9141995 0.9455508
##Solution 4  0.9034868 0.9433203
##Solution 5  0.9034868 0.9433203
##Solution 6  0.9020271 0.9428967
##Solution 7  0.9020271 0.9428967
##Solution 8  0.8988192 0.9428967
##Solution 9  0.8988192 0.9357982
##Solution 10 0.8988192 0.9357982
##
##$bestvalues
##   Card.3    Card.4
##0.9141995 0.9557145
##
##$bestsets
##       Var.1 Var.2 Var.3 Var.4
##Card.3     1     2     3     0
##Card.4     2     4     5     6
##
##$call
##genetic(mat = cor(swiss), kmin = 3, kmax = 4, popsize = 10, nger = 5,
##    criterion = "Rv")



## --------------------------------------------------------------------

##
## 2) An example of subset selection in the context of Multiple Linear
## Regression. Variable 5 (average car price) in the Cars93 MASS library
## data set is regressed on 13 other variables. The six-variable subsets
## of linear predictors are chosen using the "CCR1_2" criterion which,
## in the case of a Linear Regression, is merely  the standard Coefficient
## of Determination, R^2 (as are the other three criteria for the
## multivariate linear hypothesis, "XI_2", "TAU_2" and "ZETA_2").
##

library(MASS)
data(Cars93)
CarsHmat <- lmHmat(Cars93[,c(7:8,12:15,17:22,25)],Cars93[,5])

names(Cars93[,5,drop=FALSE])
##  [1] "Price"

colnames(CarsHmat)
```

```
##  [1] "MPG.city"           "MPG.highway"        "EngineSize"
##  [4] "Horsepower"         "RPM"                "Rev.per.mile"
##  [7] "Fuel.tank.capacity" "Passengers"         "Length"
## [10] "Wheelbase"          "Width"              "Turn.circle"
## [13] "Weight"


genetic(CarsHmat$mat, kmin=6,  H=CarsHmat$H, r=1, crit="CCR12")

##
## (Partial results only)
##
## $subsets
##             Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1      4     5     9    10    11    12
## Solution 2      4     5     9    10    11    12
## Solution 3      4     5     9    10    11    12
## Solution 4      4     5     9    10    11    12
## Solution 5      4     5     9    10    11    12
## Solution 6      4     5     9    10    11    12
## Solution 7      4     5     8    10    11    12
##
## (...)
##
## Solution 94      1     4     5     6    10    11
## Solution 95      1     4     5     6    10    11
## Solution 96      1     4     5     6    10    11
## Solution 97      1     4     5     6    10    11
## Solution 98      1     4     5     6    10    11
## Solution 99      1     4     5     6    10    11
## Solution 100     1     4     5     6    10    11
##
## $values
##   Solution 1   Solution 2   Solution 3   Solution 4   Solution 5   Solution 6
##    0.7310150    0.7310150    0.7310150    0.7310150    0.7310150    0.7310150
##   Solution 7   Solution 8   Solution 9  Solution 10  Solution 11  Solution 12
##    0.7310150    0.7271056    0.7271056    0.7271056    0.7271056    0.7271056
##  Solution 13  Solution 14  Solution 15  Solution 16  Solution 17  Solution 18
##    0.7271056    0.7270257    0.7270257    0.7270257    0.7270257    0.7270257
##
## (...)
##
##  Solution 85  Solution 86  Solution 87  Solution 88  Solution 89  Solution 90
##    0.7228800    0.7228800    0.7228800    0.7228800    0.7228800    0.7228800
##  Solution 91  Solution 92  Solution 93  Solution 94  Solution 95  Solution 96
##    0.7228463    0.7228463    0.7228463    0.7228463    0.7228463    0.7228463
##  Solution 97  Solution 98  Solution 99 Solution 100
##    0.7228463    0.7228463    0.7228463    0.7228463
##
## $bestvalues
##   Card.6
## 0.731015
##
```

```
## $bestsets
## Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
##     4     5     9    10    11    12
##
## $call
## genetic(mat = CarsHmat$mat, kmin = 6, criterion = "CCR12", H = CarsHmat$H,
##     r = 1)


## ---------------------------------------------------------------------

## 3) An example of subset selection in the context of a Canonical
## Correlation Analysis. Two groups of variables within the Cars93
## MASS library data set are compared. The goal is to select 4- to
## 6-variable subsets of the 13-variable 'X' group that are optimal in
## terms of preserving the canonical correlations, according to the
## "ZETA_2" criterion (Warning: the 3-variable 'Y' group is kept
## intact; subset selection is carried out in the 'X'
## group only).  The 'tolsym' parameter is used to relax the symmetry
## requirements on the effect matrix H which, for numerical reasons,
## is slightly asymmetric. Since corresponding off-diagonal entries of
## matrix H are different, but by less than tolsym, H is replaced
## by its symmetric part: (H+t(H))/2.

library(MASS)
data(Cars93)
CarsHmat <- lmHmat(Cars93[,c(7:8,12:15,17:22,25)],Cars93[,4:6])

names(Cars93[,4:6])
## [1] "Min.Price" "Price"     "Max.Price"

colnames(CarsHmat$mat)

##  [1] "MPG.city"           "MPG.highway"        "EngineSize"
##  [4] "Horsepower"         "RPM"                "Rev.per.mile"
##  [7] "Fuel.tank.capacity" "Passengers"         "Length"
## [10] "Wheelbase"          "Width"              "Turn.circle"
## [13] "Weight"


genetic(CarsHmat$mat, kmin=5, kmax=6, H=CarsHmat$H, r=3, crit="zeta2", tolsym=1e-9)

##  (PARTIAL RESULTS ONLY)
##
## $subsets
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     4     5     9    10    11     0
## Solution 2     4     5     9    10    11     0
## Solution 3     4     5     9    10    11     0
## Solution 4     4     5     9    10    11     0
## Solution 5     4     5     9    10    11     0
## Solution 6     4     5     9    10    11     0
```

```
## Solution 7      4     5     9    10    11     0
## Solution 8      3     4     9    10    11     0
## Solution 9      3     4     9    10    11     0
## Solution 10     3     4     9    10    11     0
##
## (...)
##
## Solution 87      3     4     6     9    10    11
## Solution 88      3     4     6     9    10    11
## Solution 89      3     4     6     9    10    11
## Solution 90      2     3     4    10    11    12
## Solution 91      2     3     4    10    11    12
## Solution 92      2     3     4    10    11    12
## Solution 93      2     3     4    10    11    12
## Solution 94      2     3     4    10    11    12
## Solution 95      2     3     4    10    11    12
## Solution 96      2     3     4    10    11    12
## Solution 97      1     3     4     6    10    11
## Solution 98      1     3     4     6    10    11
## Solution 99      1     3     4     6    10    11
## Solution 100     1     3     4     6    10    11
##
##
## $values
##
##                 card.5    card.6
## Solution 1  0.5018922 0.5168627
## Solution 2  0.5018922 0.5168627
## Solution 3  0.5018922 0.5168627
## Solution 4  0.5018922 0.5168627
## Solution 5  0.5018922 0.5168627
## Solution 6  0.5018922 0.5168627
## Solution 7  0.5018922 0.5096500
## Solution 8  0.4966191 0.5096500
## Solution 9  0.4966191 0.5096500
## Solution 10 0.4966191 0.5096500
##
## (...)
##
## Solution 87  0.4893824 0.5038649
## Solution 88  0.4893824 0.5038649
## Solution 89  0.4893824 0.5038649
## Solution 90  0.4893824 0.5035489
## Solution 91  0.4893824 0.5035489
## Solution 92  0.4893824 0.5035489
## Solution 93  0.4893824 0.5035489
## Solution 94  0.4893824 0.5035489
## Solution 95  0.4893824 0.5035489
## Solution 96  0.4893824 0.5035489
## Solution 97  0.4890986 0.5035386
## Solution 98  0.4890986 0.5035386
## Solution 99  0.4890986 0.5035386
## Solution 100 0.4890986 0.5035386
```

```
##
## $bestvalues
##    Card.5    Card.6
## 0.5018922 0.5168627
##
## $bestsets
##        Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Card.5     4     5     9    10    11     0
## Card.6     4     5     9    10    11    12
##
## $call
## genetic(mat = CarsHmat$mat, kmin = 5, kmax = 6, criterion = "zeta2",
##     H = CarsHmat$H, r = 3, tolsym = 1e-09)
##
## Warning message:
##
##  The effect description matrix (H) supplied was slightly asymmetric:
##  symmetric entries differed by up to 3.63797880709171e-12.
##  (less than the 'tolsym' parameter).
##  The H matrix has been replaced by its symmetric part.
##  in: validnovcrit(mat, criterion, H, r, p, tolval, tolsym)
##

## The selected best variable subsets

colnames(CarsHmat$mat)[c(4,5,9,10,11)]

## [1] "Horsepower" "RPM"        "Length"     "Wheelbase"  "Width"

colnames(CarsHmat$mat)[c(4,5,9,10,11,12)]

## [1] "Horsepower"  "RPM"         "Length"      "Wheelbase"   "Width"
## [6] "Turn.circle"

## --------------------------------------------------------------------
```

---

glhHmat *Total and Effect Deviation Matrices for General Linear Hypothesis*

Description

Computes total and effect matrices of Sums of Squares and Cross-Product (SSCP) deviations for a general multivariate effect characterized by the violation of a linear hypothesis. These matrices may be used as input to the variable selection search routines `anneal`, `genetic` `improve` or `eleaps`.

Usage

```
## Default S3 method:
glhHmat(x,A,C,...)
```

```
## S3 method for class 'data.frame'
glhHmat(x,A,C,...)

## S3 method for class 'formula'
glhHmat(formula,C,data=NULL,...)
```

Arguments

`x`	A matrix or data frame containing the variables for which the SSCP matrix is to be computed.
`A`	A matrix or data frame containing a design matrix specifying a linear model in which x is the response.
`C`	A matrix or vector containing the coefficients of the reference hypothesis.
`formula`	A formula of the form `'x ~ A1 + A2 + ...'` That is, the response is the set of variables whose subsets are to be compared and the right hand side specifies the columns of the design matrix.
`data`	Data frame from which variables specified in 'formula' are preferentially to be taken.
`...`	further arguments for the method.

Details

Consider a multivariate linear model $x = A\Psi + U$ and a reference hypothesis $H0 : C\Psi = 0$, with Ψ being a matrix of unknown parameters and C a known coefficient matrix with rank `r`. It is well known that, under classical Gaussian assumptions, H_0 can be tested by several increasing functions of the `r` positive eigenvalues of a product $T^{-1}H$, where T and H are total and effect matrices of SSCP deviations associated with H_0. Furthermore, whether or not the classical assumptions hold, the same eigenvalues can be used to define descriptive indices that measure an "effect" characterized by the violation of H_0 (see reference [1] for further details). Those SSCP matrices are given by $T = x'(I - P_\omega)x$ and $H = x'(P_\Omega - P_\omega)x$, where I is an identity matrix and $P_\Omega = A(A'A)^- A'$,

$$P_\omega = A(A'A)^- A' - A(A'A)^- C'[C(A'A)^- C']^- C(A'A)^- A'$$

are projection matrices on the spaces spanned by the columns of A (space Ω) and by the linear combinations of these columns that satisfy the reference hypothesis (space ω). In these formulae M' denotes the transpose of M and M^- a generalized inverse. `glhHmat` computes the T and H matrices which then can be used as input to the search routines `anneal`, `genetic` `improve` and `eleaps` that try to select subsets of x according to their contribution to the violation of H_0.

Value

A list with four items:

`mat`	The total SSCP matrix
`H`	The effect SSCP matrix

`r` The expected rank of the H matrix which equals the rank of C. The true rank of H can be different from r if the x variables are linearly dependent.

`call` The function call which generated the output.

References

[1] Duarte Silva. A.P. (2001). Efficient Variable Screening for Multivariate Analysis, *Journal of Multivariate Analysis*, Vol. 76, 35-62.

See Also

`anneal`, `genetic`, `improve`, `eleaps`, `lmHmat`, `ldaHmat`.

Examples

```
##---------------------------------------------------------------------------

## The following examples create T and H matrices for different analysis
## of the MASS data set "crabs". This data records physical measurements
## on 200 specimens of Leptograpsus variegatus crabs observed on the shores
## of Western Australia. The crabs are classified by two factors, sex and sp
## (crab species as defined by its colour: blue or orange), with two levels
## each. The measurement variables include the carapace length (CL),
## the carapace width (CW), the size of the frontal lobe (FL) and the size of
## the rear width (RW). In the  analysis provided, we assume that there is
## an interest in comparing the subsets of these variables measured in their
## original and logarithmic scales.

library(MASS)
data(crabs)
lFL <- log(crabs$FL)
lRW <- log(crabs$RW)
lCL <- log(crabs$CL)
lCW <- log(crabs$CW)

# 1)  Create the T and H matrices associated with a linear
# discriminant analysis on the groups defined by the sp factor.
# This call is equivalent to ldaHmat(sp ~ FL + RW + CL + CW  + lFL +
# lRW + lCL + lCW,crabs)

Hmat1 <- glhHmat(cbind(FL,RW,CL,CW,lFL,lRW,lCL,lCW) ~ sp,c(0,1),crabs)
Hmat1

##$mat
##            FL        RW         CL         CW        lFL        lRW        lCL
##FL   2431.2422 1623.4509  4846.9787  5283.6093 162.718609 133.360397 158.865134
##RW   1623.4509 1317.7935  3254.5776  3629.6883 109.877182 107.287243 108.335721
##CL   4846.9787 3254.5776 10085.3040 11096.5141 326.243285 269.564742 330.912570
##CW   5283.6093 3629.6883 11096.5141 12331.5680 356.317934 300.786770 364.620761
##lFL   162.7186  109.8772   326.2433   356.3179  11.114733   9.188391  10.910730
##lRW   133.3604  107.2872   269.5647   300.7868   9.188391   8.906350   9.130692
##lCL   158.8651  108.3357   330.9126   364.6208  10.910730   9.130692  11.088706
```

```
##lCW  152.7872  106.4277   321.0253   357.0051  10.503303   8.970570  10.765175
##          lCW
##FL  152.78716
##RW  106.42775
##CL  321.02534
##CW  357.00510
##lFL  10.50330
##lRW   8.97057
##lCL  10.76517
##lCW  10.54334

##$H
##           FL         RW        CL        CW        lFL        lRW        lCL
##FL  466.34580 247.526700 625.30650 518.41650 30.7408809 19.4543206 20.5494907
##RW  247.52670 131.382050 331.89975 275.16475 16.3166234 10.3259508 10.9072444
##CL  625.30650 331.899750 838.45125 695.12625 41.2193540 26.0856066 27.5540813
##CW  518.41650 275.164750 695.12625 576.30125 34.1733106 21.6265286 22.8439819
##lFL  30.74088  16.316623  41.21935  34.17331  2.0263971  1.2824024  1.3545945
##lRW  19.45432  10.325951  26.08561  21.62653  1.2824024  0.8115664  0.8572531
##lCL  20.54949  10.907244  27.55408  22.84398  1.3545945  0.8572531  0.9055117
##lCW  15.16136   8.047335  20.32933  16.85423  0.9994161  0.6324790  0.6680840
##          lCW
##FL  15.1613582
##RW   8.0473352
##CL  20.3293260
##CW  16.8542276
##lFL  0.9994161
##lRW  0.6324790
##lCL  0.6680840
##lCW  0.4929106

##$r
##[1] 1

##$call
##glhHmat.formula(formula = cbind(FL, RW, CL, CW, lFL, lRW, lCL,
##    lCW) ~ sp, C = c(0, 1), data = crabs)


# 2)  Create the T and H matrices associated with an analysis
# of the interactions between the sp and sex factors

Hmat2 <- glhHmat(cbind(FL,RW,CL,CW,lFL,lRW,lCL,lCW) ~ sp*sex,c(0,0,0,1),crabs)
Hmat2

##$mat
##           FL         RW         CL         CW        lFL        lRW        lCL
##FL  1960.3362 1398.52890  4199.1581  4747.5409 131.651804 115.607172 137.663744
##RW  1398.5289 1074.36105  3034.2793  3442.0233  95.176151  88.529040 100.659912
##CL  4199.1581 3034.27925  9135.6987 10314.2389 283.414814 251.877591 300.140005
##CW  4747.5409 3442.02325 10314.2389 11686.9387 320.883015 285.744945 339.253367
##lFL  131.6518   95.17615   283.4148   320.8830   9.065041   8.027569   9.509543
##lRW  115.6072   88.52904   251.8776   285.7449   8.027569   7.460222   8.516618
```

```
##lCL  137.6637  100.65991   300.1400   339.2534   9.509543   8.516618  10.090003
##lCW  137.2059  100.46203   298.6227   338.5254   9.473873   8.494741  10.037059
##           lCW
##FL  137.205863
##RW  100.462028
##CL  298.622747
##CW  338.525352
##lFL   9.473873
##lRW   8.494741
##lCL  10.037059
##lCW  10.011755

##$H
##            FL          RW        CL         CW        lFL        lRW        lCL
##FL   80.645000  68.389500 153.73350 191.57950  5.4708199  5.1596883  5.2140868
##RW   68.389500  57.996450 130.37085 162.46545  4.6394276  4.3755782  4.4217098
##CL  153.733500 130.370850 293.06205 365.20785 10.4290197  9.8359098  9.9396095
##CW  191.579500 162.465450 365.20785 455.11445 12.9964281 12.2573068 12.3865353
##lFL   5.470820   4.639428  10.42902  12.99643  0.3711311  0.3500245  0.3537148
##lRW   5.159688   4.375578   9.83591  12.25731  0.3500245  0.3301182  0.3335986
##lCL   5.214087   4.421710   9.93961  12.38654  0.3537148  0.3335986  0.3371158
##lCW   5.584150   4.735535  10.64506  13.26565  0.3788193  0.3572754  0.3610421
##           lCW
##FL   5.5841501
##RW   4.7355352
##CL  10.6450610
##CW  13.2656543
##lFL  0.3788193
##lRW  0.3572754
##lCL  0.3610421
##lCW  0.3866667

##$r
##[1] 1

##$call
##glhHmat.formula(formula = cbind(FL, RW, CL, CW, lFL, lRW, lCL,
##    lCW) ~ sp * sex, C = c(0, 0, 0, 1), data = crabs)

## 3)  Create the T and H matrices associated with an analysis
## of the effect of the sp factor after controlling for sex

C <- matrix(0.,2,4)
C[1,3] = C[2,4] = 1.
C

##       [,1] [,2] [,3] [,4]
##  [1,]    0    0    1    0
##  [2,]    0    0    0    1

Hmat3 <- glhHmat(cbind(FL,RW,CL,CW,lFL,lRW,lCL,lCW) ~ sp*sex,C,crabs)
Hmat3
```

```
##$mat
##          FL         RW         CL         CW        lFL        lRW        lCL
##FL  1964.8964 1375.92420  4221.6722  4765.1928 131.977728 113.906076 138.315643
##RW  1375.9242 1186.41150  2922.6779  3354.5236  93.560559  96.961292  97.428477
##CL  4221.6722 2922.67790  9246.8527 10401.3878 285.023931 243.479136 303.358489
##CW  4765.1928 3354.52360 10401.3878 11755.2667 322.144623 279.160241 341.776779
##lFL  131.9777   93.56056   285.0239   322.1446   9.088336   7.905989   9.556135
##lRW  113.9061   96.96129   243.4791   279.1602   7.905989   8.094783   8.273439
##lCL  138.3156   97.42848   303.3585   341.7768   9.556135   8.273439  10.183194
##lCW  137.6258   98.38041   300.6960   340.1509   9.503886   8.338091  10.097091
##           lCW
##FL  137.625801
##RW   98.380414
##CL  300.696018
##CW  340.150874
##lFL   9.503886
##lRW   8.338091
##lCL  10.097091
##lCW  10.050426

##$H
##           FL         RW         CL         CW        lFL         lRW
##FL   85.205200  45.784800 176.247600 209.231400  5.7967443  3.45859277
##RW   45.784800 170.046900  18.769500  74.965800  3.0238356 12.80782993
##CL  176.247600  18.769500 404.216100 452.356800 12.0381364  1.43745463
##CW  209.231400  74.965800 452.356800 523.442500 14.2580360  5.67260253
##lFL   5.796744   3.023836  12.038136  14.258036  0.3944254  0.22844463
##lRW   3.458593  12.807830   1.437455   5.672603  0.2284446  0.96467943
##lCL   5.865986   1.190274  13.158093  14.909948  0.4003070  0.09041999
##lCW   6.004088   2.653921  12.718332  14.891177  0.4088329  0.20062548
##           lCL        lCW
##FL   5.86598627  6.0040883
##RW   1.19027431  2.6539211
##CL  13.15809339 12.7183319
##CW  14.90994753 14.8911765
##lFL  0.40030704  0.4088329
##lRW  0.09041999  0.2006255
##lCL  0.43030750  0.4210740
##lCW  0.42107404  0.4253378

##$r
##[1] 2

##$call
##glhHmat.formula(formula = cbind(FL, RW, CL, CW, lFL, lRW, lCL,
##    lCW) ~ sp * sex, C = C, data = crabs)
```

`glmHmat` *Input matrices for subselect search routines in generalized linear models*

Description

glmHmat uses a glm object (fitdglmmodel) to build an estimate of Fisher's Information (FI) matrix together with an auxiliarly rank-one positive-defenite matrix (H), such that the positive eigenvalue of $FI^{-1}H$ equals the value of Wald's statistic for testing the global significance of fitdglmmodel. These matrices may be used as input to the variable selection search routines `anneal`, `genetic` `improve` or `eleaps`, usign the minimization of Wald's statistic as criterion for discarding variables.

Usage

```
## S3 method for class 'glm'
glmHmat(fitdglmmodel,...)
```

Arguments

`fitdglmmodel`	A glm object containaing the estimates, and respective covariance matrix, of a generalized linear model.
`...`	further arguments for the method.

Details

Variable selection in the context of generalized linear models is typically based on the minimization of statistics that test the significance of excluded variables. In particular, the likelihood ratio, Wald's, Rao's and some adaptations of such statistics, are often proposed as comparison criteria for variable subsets of the same dimensionality. All these statistics are assympotically equivalent and can be converted into information criteria, like the AIC, that are also able to compare subsets of different dimensionalities (see references [1] and [2] for further details).

Among these criteria, Wald's statistic has some computational advantages because it can always be derived from the same (concerning the full model) maximum likelihood and Fisher information estimates. In particular, if W_{allv} is the value of the Wald statistic testing the significance of the full covariate vector, b and FI are coefficient and Fisher information estimates and H is an auxiliary rank-one matrix given by `H = FI %*% b %*% t(b) %*% FI`, it follows that the value of Wald's statistic for the excluded variables (W_excv) in a given subset is given by $W_{excv} = W_{allv} - tr(FI_{indices}^{-1} H_{indices})$, where $FI_indices$ and $H_indices$ are the portions of the FI and H matrices associated with the selected variables.

glmHmat retrieves the values of the FI and H matrices from a glm object. These matrices may then be used as input to the search functions `anneal`, `genetic`, `improve` and `eleaps`.

Value

A list with four items:

`mat`	An estimate (FI) of Fisher's information matrix for the full model variable-coefficient estimates
`H`	A product of the form (`FI %*% b %*% t(b) %*% FI`) where b is a vector of variable-coefficient estimates
`r`	The rank of the H matrix. Always set to one in glmHmat.
`call`	The function call which generated the output.

References

[1] Lawless, J. and Singhal, K. (1978). Efficient Screening of Nonnormal Regression Models, *Biometrics*, Vol. 34, 318-327.

[2] Lawless, J. and Singhal, K. (1987). ISMOD: An All-Subsets Regression Program for Generalized Models I. Statistical and Computational Background, *Computer Methods and Programs in Biomedicine*, Vol. 24, 117-124.

See Also

anneal, genetic, improve, eleaps, glm.

Examples

```
##-----------------------------------------------------------------

##-----------------------------------------------------------------

##  An example of variable selection in the context of binary response
##  regression models. We consider the last 100 observations of
##  the iris data set (versicolor an verginica species) and try
##  to find the best variable subsets for models that take species
##  as the response variable.

data(iris)
iris2sp <- iris[iris$Species != "setosa",]

# Create the input matrices for the  search routines in a logistic regression model

modelfit <- glm(Species ~ Sepal.Length + Sepal.Width + Petal.Length +
Petal.Width,iris2sp,family=binomial)
Hmat <- glmHmat(modelfit)
Hmat

## $mat
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length  0.28340358 0.03263437  0.09552821  -0.01779067
## Sepal.Width   0.03263437 0.13941541  0.01086596 0.04759284
## Petal.Length  0.09552821 0.01086596  0.08847655  -0.01853044
## Petal.Width   -0.01779067 0.04759284  -0.01853044 0.03258730

## $H
##              Sepal.Length  Sepal.Width Petal.Length  Petal.Width
## Sepal.Length   0.11643732  0.013349227 -0.063924853 -0.050181400
## Sepal.Width    0.01334923  0.001530453 -0.007328813 -0.005753163
## Petal.Length  -0.06392485 -0.007328813  0.035095164  0.027549918
## Petal.Width   -0.05018140 -0.005753163  0.027549918  0.021626854

## $r
## [1] 1

## $call
```

```
## glmHmat(fitdglmmodel = modelfit)

# Search for the 3 best variable subsets of each dimensionality by an exausitve search

eleaps(Hmat$mat,H=Hmat$H,r=1,criterion="Wald",nsol=3)

## $subsets
## , , Card.1

##            Var.1 Var.2 Var.3
## Solution 1     4     0     0
## Solution 2     1     0     0
## Solution 3     3     0     0

## , , Card.2

##           Var.1 Var.2 Var.3
## Solution 1     1     3     0
## Solution 2     3     4     0
## Solution 3     2     4     0

## , , Card.3

##            Var.1 Var.2 Var.3
## Solution 1     2     3     4
## Solution 2     1     3     4
## Solution 3     1     2     3


## $values
##              card.1   card.2   card.3
## Solution 1 4.894554 3.522885 1.060121
## Solution 2 5.147360 3.952538 2.224335
## Solution 3 5.161553 3.972410 3.522879

## $bestvalues
##   Card.1   Card.2   Card.3
## 4.894554 3.522885 1.060121

## $bestsets
##        Var.1 Var.2 Var.3
## Card.1     4     0     0
## Card.2     1     3     0
## Card.3     2     3     4

## $call
## eleaps(mat = Hmat$mat, nsol = 3, criterion = "Wald", H = Hmat$H,
##    r = 1)


## It should be stressed that, unlike other criteria in the
## subselect package, the Wald criterion is not bounded above by
## 1 and is a decreasing function of subset quality, so that the
```

```
## 3-variable subsets do, in fact, perform better than their smaller-sized
## counterparts.


## >
## > proc.time()
## [1] 0.680 0.064 0.736 0.000 0.000
```

---

improve *Restricted Local Improvement search for an optimal k-variable subset*

Description

Given a set of variables, a Restricted Local Improvement algorithm seeks a k-variable subset which is optimal, as a surrogate for the whole set, with respect to a given criterion.

Usage

```
improve( mat, kmin, kmax = kmin, nsol = 1, exclude = NULL,
include = NULL, setseed = FALSE, criterion = "default", pcindices="first_k",
initialsol = NULL, force = FALSE, H=NULL, r=0,
tolval=1000*.Machine$double.eps,tolsym=1000*.Machine$double.eps)
```

Arguments

`mat`	a covariance/correlation, information or sums of squares and products matrix of the variables from which the k-subset is to be selected. See the `Details` section below.
`kmin`	the cardinality of the smallest subset that is wanted.
`kmax`	the cardinality of the largest subset that is wanted.
`nsol`	the number of different subsets (runs of the algorithm) wanted.
`exclude`	a vector of variables (referenced by their row/column numbers in matrix `mat`) that are to be forcibly excluded from the subsets.
`include`	a vector of variables (referenced by their row/column numbers in matrix `mat`) that are to be forcibly included from the subsets.
`setseed`	logical variable indicating whether to fix an initial seed for the random number generator, which will be re-used in future calls to this function whenever setseed is again set to TRUE.
`criterion`	Character variable, which indicates which criterion is to be used in judging the quality of the subsets. Currently, the "Rm", "Rv", "Gcd", "Tau2", "Xi2", "Zeta2", "ccr12" and "Wald" criteria are supported (see the `Details` section, the `References` and the links `rm.coef`, `rv.coef`, `gcd.coef`, `tau2.coef`, `xi2.coef`, `zeta2.coef` and `ccr12.coef` for further details). The default criterion is "Rm" if parameter `r` is zero (exploratory and PCA problems), "Wald" if `r` is equal to one and `mat` has a "FisherI" attribute set to TRUE (generalized linear models), and "Tau2" otherwise (multivariate linear model framework).

pcindices
: either a vector of ranks of Principal Components that are to be used for comparison with the k-variable subsets (for the Gcd criterion only, see `gcd.coef`) or the default text `first_k`. The latter will associate PCs 1 to *k* with each cardinality *k* that has been requested by the user.

initialsol
: vector, matrix or 3-d array of initial solutions for the restricted local improvement search. If a *single cardinality* is required, `initialsol` may be a vector of length *k*(accepted even if `nsol` > 1, in which case it is used as the initial solution for all `nsol` final solutions that are requested with a warning that the same initial solution necessarily produces the same final solution); a 1 x *k* matrix (as produced by the `$bestsets` output value of the algorithm functions `anneal`, `genetic`, or `improve`), or a 1 x *k* x 1 array (as produced by the `$subsets` output value), in which case it will be treated as the above k-vector; or an `nsol x k` matrix, or `nsol` x `k` x 1 3-d array, in which case each row (dimension 1) will be used as the initial solution for each of the `nsol` final solutions requested. If *more than one cardinality* is requested, `initialsol` can be a `length(kmin:kmax) x kmax` matrix (as produced by the `$bestsets` option of the algorithm functions) (even if `nsol` > 1, in which case each row will be replicated to produced the initial solution for all `nsol` final solutions requested in each cardinality, with a warning that a single initial solution necessarily produces identical final solutions), or a `nsol x kmax x length(kmin:kmax)` 3-d array (as produced by the `$subsets` output option), in which case each row (dimension 1) is interpreted as a different initial solution.

  If the `exclude` and/or `include` options are used, `initialsol` must also respect those requirements.

force
: a logical variable indicating whether, for large data sets (currently `p` > 400) the algorithm should proceed anyways, regardless of possible memory problems which may crash the R session.

H
: Effect description matrix. Not used with the Rm, Rv or Gcd criteria, hence the NULL default value. See the `Details` section below.

r
: Expected rank of the effects (`H`) matrix. Not used with the Rm, Rv or Gcd criteria. See the `Details` section below.

tolval
: the tolerance level for the reciprocal of the 2-norm condition number of the correlation/covariance matrix, i.e., for the ratio of the smallest to the largest eigenvalue of the input matrix. Matrices with a reciprocal of the condition number smaller than `tolval` will activate a restricted-search for well conditioned subsets.

tolsym
: the tolerance level for symmetry of the covariance/correlation/total matrix and for the effects (`H`) matrix. If corresponding matrix entries differ by more than this value, the input matrices will be considered asymmetric and execution will be aborted. If corresponding entries are different, but by less than this value, the input matrix will be replaced by its symmetric part, i.e., input matrix A becomes (A+t(A))/2.

Details

An initial `k`-variable subset (for `k` ranging from `kmin` to `kmax`) of a full set of `p` variables is randomly selected and the variables not belonging to this subset are placed in a queue. The possibility of replacing a variable in the current k-subset with a variable from the queue is then explored. More precisely, a variable is selected, removed from the queue, and the k values of the criterion which would result from swapping this selected variable with each variable in the current subset are computed. If the best of these values improves the current criterion value, the current subset is updated accordingly. In this case, the variable which leaves the subset is added to the queue, but only if it has not previously been in the queue (i.e., no variable can enter the queue twice). The algorithm proceeds until the queue is emptied.

The user may force variables to be included and/or excluded from the k-subsets, and may specify initial solutions.

For each cardinality k, the total number of calls to the procedure which computes the criterion values is O(`nsol` x k x p). These calls are the dominant computational effort in each iteration of the algorithm.

In order to improve computation times, the bulk of computations are carried out in a Fortran routine. Further details about the algorithm can be found in Reference 1 and in the comments to the Fortran code (in the `src` subdirectory for this package). For datasets with a very large number of variables (currently p > 400), it is necessary to set the `force` argument to TRUE for the function to run, but this may cause a session crash if there is not enough memory available.

The function checks for ill-conditioning of the input matrix (specifically, it checks whether the ratio of the input matrix's smallest and largest eigenvalues is less than `tolval`). For an ill-conditioned input matrix, the search is restricted to its well-conditioned subsets. The function `trim.matrix` may be used to obtain a well-conditioned input matrix.

In a general descriptive (Principal Components Analysis) setting, the three criteria Rm, Rv and Gcd can be used to select good k-variable subsets. Arguments `H` and `r` are not used in this context. See references [1] and [2] and the `Examples` for a more detailed discussion.

In the setting of a multivariate linear model, $X = A\Psi + U$, criteria Ccr12, Tau2, Xi2 and Zeta2 can be used to select subsets according to their contribution to an effect characterized by the violation of a reference hypothesis, $C\Psi = 0$ (see reference [3] for further details). In this setting, arguments `mat` and `H` should be set respectively to the usual Total (Hypothesis + Error) and Hypothesis, Sum of Squares and Cross-Products (SSCP) matrices. Argument `r` should be set to the expected rank of `H`. Currently, for reasons of computational efficiency, criterion Ccr12 is available only when `r` $\leq$ 3. Particular cases in this setting include Linear Discriminant Analysis (LDA), Linear Regression Analysis (LRA), Canonical Correlation Analysis (CCA) with one set of variables fixed and several extensions of these and other classical multivariate methodologies.

In the setting of a generalized linear model, criterion Wald can be used to select subsets according to the (lack of) significance of the discarded variables, as measured by the respective Wald's statistic (see reference [4] for further details). In this setting arguments `mat` and `H` should be set respectively to FI and `FI %*% b %*% t(b) %*% FI`, where `b` is a column vector of variable coefficient estimates and FI is an estimate of the corresponding Fisher information matrix.

The auxiliary functions `lmHmat`, `ldaHmat` `glhHmat` and `glmHmat` are provided to automatically create the matrices `mat` and `H` in all the cases considered.

Value

A list with five items:

`subsets` An `nsol` x `kmax` x length(`kmin`:`kmax`) 3-dimensional array, giving for each cardinality (dimension 3) and each solution (dimension 1) the list of variables (referenced by their row/column numbers in matrix mat) in the subset (dimension 2). (For cardinalities smaller than kmax, the extra final positions are set to zero).

values — An nsol x length(kmin:kmax) matrix, giving for each cardinality (columns), the criterion values of the nsol (rows) solutions obtained.

bestvalues — A length(kmin:kmax) vector giving the best values of the criterion obtained for each cardinality.

bestsets — A length(kmin:kmax) x kmax matrix, giving, for each cardinality (rows), the variables (referenced by their row/column numbers in matrix mat) in the best k-subset that was found.

call — The function call which generated the output.

References

[1] Cadima, J., Cerdeira, J. Orestes and Minhoto, M. (2004) Computational aspects of algorithms for variable selection in the context of principal components. *Computational Statistics and Data Analysis*, 47, 225-236.

[2]Cadima, J. and Jolliffe, I.T. (2001). Variable Selection and the Interpretation of Principal Subspaces, *Journal of Agricultural, Biological and Environmental Statistics*, Vol. 6, 62-79.

[3]Duarte Silva, A.P. (2001) Efficient Variable Screening for Multivariate Analysis, *Journal of Multivariate Analysis*, Vol. 76, 35-62.

[4] Lawless, J. and Singhal, K. (1978). Efficient Screening of Nonnormal Regression Models, *Biometrics*, Vol. 34, 318-327.

See Also

rm.coef, rv.coef, gcd.coef, tau2.coef, xi2.coef, zeta2.coef, ccr12.coef, genetic, anneal, eleaps, trim.matrix, lmHmat, ldaHmat, glhHmat, glmHmat.

Examples

```
## --------------------------------------------------------------------

##
## 1) For illustration of use, a small data set with very few iterations
## of the algorithm.
## Subsets of 2 and of 3 variables are sought using the RM criterion.
##

data(swiss)
improve(cor(swiss),2,3,nsol=4,criterion="GCD")
## $subsets
## , , Card.2
##
##            Var.1 Var.2 Var.3
## Solution 1     3     6     0
## Solution 2     3     6     0
## Solution 3     3     6     0
## Solution 4     3     6     0
##
```

```
## , , Card.3
##
##            Var.1 Var.2 Var.3
## Solution 1     4     5     6
## Solution 2     4     5     6
## Solution 3     4     5     6
## Solution 4     4     5     6
##
##
## $values
##               card.2   card.3
## Solution 1 0.8487026 0.925372
## Solution 2 0.8487026 0.925372
## Solution 3 0.8487026 0.925372
## Solution 4 0.8487026 0.925372
##
## $bestvalues
##    Card.2    Card.3
## 0.8487026 0.9253720
##
## $bestsets
##        Var.1 Var.2 Var.3
## Card.2     3     6     0
## Card.3     4     5     6
##
##$call
##improve(cor(swiss), 2, 3, nsol = 4, criterion = "GCD")


## ---------------------------------------------------------------------

##
## 2) Forcing the inclusion of variable 1 in the subset
##

 improve(cor(swiss),2,3,nsol=4,criterion="GCD",include=c(1))

## $subsets
## , , Card.2
##
##            Var.1 Var.2 Var.3
## Solution 1     1     6     0
## Solution 2     1     6     0
## Solution 3     1     6     0
## Solution 4     1     6     0
##
## , , Card.3
##
##            Var.1 Var.2 Var.3
## Solution 1     1     5     6
## Solution 2     1     5     6
## Solution 3     1     5     6
## Solution 4     1     5     6
```

```
##
##
## $values
##               card.2    card.3
## Solution 1 0.7284477 0.8048528
## Solution 2 0.7284477 0.8048528
## Solution 3 0.7284477 0.8048528
## Solution 4 0.7284477 0.8048528
##
## $bestvalues
##    Card.2    Card.3
## 0.7284477 0.8048528
##
## $bestsets
##        Var.1 Var.2 Var.3
## Card.2     1     6     0
## Card.3     1     5     6
##
##$call
##improve(cor(swiss), 2, 3, nsol = 4, criterion = "GCD", include = c(1))

## --------------------------------------------------------------------

## 3) An example of subset selection in the context of Multiple Linear
## Regression. Variable 5 (average car price) in the Cars93 MASS library
## data set is regressed on 13 other variables. Three variable subsets of
## cardinalities 4, 5 and 6 are requested, using the "XI_2" criterion which,
## in the case of a Linear Regression, is merely  the standard Coefficient of
## Determination, R^2 (as are the other three criteria for the
## multivariate linear hypothesis, "TAU_2", "CCR1_2" and "ZETA_2").

library(MASS)
data(Cars93)
CarsHmat <- lmHmat(Cars93[,c(7:8,12:15,17:22,25)],Cars93[,5])

names(Cars93[,5,drop=FALSE])
##  [1] "Price"

colnames(CarsHmat$mat)

##  [1] "MPG.city"           "MPG.highway"        "EngineSize"
##  [4] "Horsepower"         "RPM"                "Rev.per.mile"
##  [7] "Fuel.tank.capacity" "Passengers"         "Length"
## [10] "Wheelbase"          "Width"              "Turn.circle"
## [13] "Weight"


improve(CarsHmat$mat, kmin=4, kmax=6, H=CarsHmat$H, r=1, crit="xi2", nsol=3)

## $subsets
## , , Card.4
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
```

```
## Solution 1     3     4    11    13     0     0
## Solution 2     3     4    11    13     0     0
## Solution 3     4     5    10    11     0     0
##
## , , Card.5
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     3     4     8    11    13     0
## Solution 2     4     5    10    11    12     0
## Solution 3     4     5    10    11    12     0
##
## , , Card.6
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     4     5     6    10    11    12
## Solution 2     4     5     8    10    11    12
## Solution 3     4     5     9    10    11    12
##
##
## $values
##               card.4    card.5    card.6
## Solution 1 0.6880773 0.6899182 0.7270257
## Solution 2 0.6880773 0.7241457 0.7271056
## Solution 3 0.7143794 0.7241457 0.7310150
##
## $bestvalues
##    Card.4    Card.5    Card.6
## 0.7143794 0.7241457 0.7310150
##
## $bestsets
##        Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Card.4     4     5    10    11     0     0
## Card.5     4     5    10    11    12     0
## Card.6     4     5     9    10    11    12
##
## $call
## improve(mat = CarsHmat$mat, kmin = 4, kmax = 6, nsol = 3, criterion = "xi2",
##     H = CarsHmat$H, r = 1)



## ----------------------------------------------------------------------

## 4) A Linear Discriminant Analysis example with a very small data set.
## We consider the Iris data and three groups, defined by species (setosa,
## versicolor and virginica). The goal is to select the 2- and 3-variable
## subsets that are optimal for the linear discrimination (as measured
## by the "TAU_2" criterion).

data(iris)
irisHmat <- ldaHmat(iris[1:4],iris$Species)
improve(irisHmat$mat,kmin=2,kmax=3,H=irisHmat$H,r=2,crit="ccr12")
```

```
## $subsets
## , , Card.2
##
##            Var.1 Var.2 Var.3
## Solution 1     2     3     0
##
## , , Card.3
##
##            Var.1 Var.2 Var.3
## Solution 1     2     3     4
##
##
## $values
##               card.2    card.3
## Solution 1 0.8079476 0.8419635
##
## $bestvalues
##    Card.2    Card.3
## 0.8079476 0.8419635
##
## $bestsets
##        Var.1 Var.2 Var.3
## Card.2     2     3     0
## Card.3     2     3     4
##
## $call
## improve(mat = irisHmat$mat, kmin = 2, kmax = 3,
##     criterion = "tau2", H = irisHmat$H, r = 2)
##

## ----------------------------------------------------------------------

## 5) An example of subset selection in the context of a Canonical
## Correlation Analysis. Two groups of variables within the Cars93
## MASS library data set are compared. The goal is to select 4- to
## 6-variable subsets of the 13-variable 'X' group that are optimal in
## terms of preserving the canonical correlations, according to the
## "ZETA_2" criterion (Warning: the 3-variable 'Y' group is kept
## intact; subset selection is carried out in the 'X'
## group only).  The 'tolsym' parameter is used to relax the symmetry
## requirements on the effect matrix H which, for numerical reasons,
## is slightly asymmetric. Since corresponding off-diagonal entries of
## matrix H are different, but by less than tolsym, H is replaced
## by its symmetric part: (H+t(H))/2.

library(MASS)
data(Cars93)
CarsHmat <- lmHmat(Cars93[,c(7:8,12:15,17:22,25)],Cars93[,4:6])

names(Cars93[,4:6])
## [1] "Min.Price" "Price"     "Max.Price"

colnames(CarsHmat$mat)
```

```
##  [1] "MPG.city"           "MPG.highway"        "EngineSize"
##  [4] "Horsepower"         "RPM"                "Rev.per.mile"
##  [7] "Fuel.tank.capacity" "Passengers"         "Length"
## [10] "Wheelbase"          "Width"              "Turn.circle"
## [13] "Weight"
```

```
improve(CarsHmat$mat, kmin=4, kmax=6, H=CarsHmat$H, r=3, crit="zeta2", tolsym=1e-9)
```

```
## $subsets
## , , Card.4
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     3     4    11    13     0     0
##
## , , Card.5
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     3     4     9    11    13     0
##
## , , Card.6
##
##            Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Solution 1     3     4     5     9    10    11
##
##
## $values
##               card.4    card.5    card.6
## Solution 1 0.4626035 0.4875495 0.5071096
##
## $bestvalues
##    Card.4    Card.5    Card.6
## 0.4626035 0.4875495 0.5071096
##
## $bestsets
##        Var.1 Var.2 Var.3 Var.4 Var.5 Var.6
## Card.4     3     4    11    13     0     0
## Card.5     3     4     9    11    13     0
## Card.6     3     4     5     9    10    11
##
## $call
## improve(mat = CarsHmat$mat, kmin = 4, kmax = 6, criterion = "zeta2",
##     H = CarsHmat$H, r = 3, tolsym = 1e-09)
##
## Warning message:
##
##  The effect description matrix (H) supplied was slightly asymmetric:
##  symmetric entries differed by up to 3.63797880709171e-12.
##  (less than the 'tolsym' parameter).
##  The H matrix has been replaced by its symmetric part.
##  in: validnovcrit(mat, criterion, H, r, p, tolval, tolsym)
```

```
## ---------------------------------------------------------------------

##  6) An example of variable selection in the context of a logistic
##  regression model. We consider the last 100 observations of
##  the iris data set (versicolor and verginica species) and try
##  to find the best variable subsets for the model that takes species
##  as response variable.

data(iris)
iris2sp <- iris[iris$Species != "setosa",]
logrfit <- glm(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
iris2sp,family=binomial)
Hmat <- glmHmat(logrfit)
improve(Hmat$mat,1,3,H=Hmat$H,r=1,criterion="Wald")

## $subsets
## , , Card.1
##
##             Var.1 Var.2 Var.3
## Solution 1     4     0     0

## , , Card.2

##             Var.1 Var.2 Var.3
## Solution 1     1     3     0

## , , Card.3

##             Var.1 Var.2 Var.3
## Solution 1     2     3     4


## $values
##               card.1   card.2   card.3
## Solution 1 4.894554 3.522885 1.060121

## $bestvalues
##   Card.1   Card.2   Card.3
## 4.894554 3.522885 1.060121

## $bestsets
##        Var.1 Var.2 Var.3
## Card.1     4     0     0
## Card.2     1     3     0
## Card.3     2     3     4

## $call
## improve(mat = Hmat$mat, kmin = 1, kmax = 3, criterion = "Wald",
##     H = Hmat$H, r = 1)
## ---------------------------------------------------------------------

## It should be stressed that, unlike other criteria in the
## subselect package, the Wald criterion is not bounded above by
```

```
## 1 and is a decreasing function of subset quality, so that the
## 3-variable subsets do, in fact, perform better than their smaller-sized
## counterparts.
```

`ldaHmat` *Total and Between-Group Deviation Matrices in Linear Discriminant Analysis*

Description

Computes total and between-group matrices of Sums of Squares and Cross-Product (SSCP) deviations in linear discriminant analysis. These matrices may be used as input to the variable selection search routines `anneal`, `genetic` `improve` or `eleaps`.

Usage

```
## Default S3 method:
ldaHmat(x,grouping,...)

## S3 method for class 'data.frame'
ldaHmat(x,grouping,...)

## S3 method for class 'formula'
ldaHmat(formula,data=NULL,...)
```

Arguments

`x`	A matrix or data frame containing the discriminators for which the SSCP matrix is to be computed.
`grouping`	A factor specifying the class for each observation.
`formula`	A formula of the form `'groups ~ x1 + x2 + ...'` That is, the response is the grouping factor and the right hand side specifies the (non-factor) discriminators.
`data`	Data frame from which variables specified in 'formula' are preferentially to be taken.
`...`	further arguments for the method.

Value

A list with four items:

`mat`	The total SSCP matrix
`H`	The between-groups SSCP matrix
`r`	The expected rank of the H matrix which equals the minimum between the number of discriminators and the number of groups minus one. The true rank of H can be different from r if the discriminators are linearly dependent.
`call`	The function call which generated the output.

See Also

anneal, genetic, improve, eleaps.

Examples

```
##-----------------------------------------------------------------------

## An example with a very small data set. We consider the Iris data
## and three groups, defined by species (setosa, versicolor and
## virginica).

data(iris)
irisHmat <- ldaHmat(iris[1:4],iris$Species)
irisHmat

##$mat
##             Sepal.Length Sepal.Width Petal.Length Petal.Width
##Sepal.Length   102.168333   -6.322667     189.8730    76.92433
##Sepal.Width     -6.322667   28.306933     -49.1188   -18.12427
##Petal.Length   189.873000  -49.118800     464.3254   193.04580
##Petal.Width     76.924333  -18.124267     193.0458    86.56993

##$H
##             Sepal.Length Sepal.Width Petal.Length Petal.Width
##Sepal.Length     63.21213   -19.95267     165.2484    71.27933
##Sepal.Width     -19.95267    11.34493     -57.2396   -22.93267
##Petal.Length    165.24840   -57.23960     437.1028   186.77400
##Petal.Width      71.27933   -22.93267     186.7740    80.41333

##$r
##[1] 2

##$call
##ldaHmat.data.frame(x = iris[1:4], grouping = iris$Species)
```

---

lmHmat *Total and Effect Deviation Matrices for Linear Regression and Canonical Correlation Analysis*

Description

Computes total an effect matrices of Sums of Squares and Cross-Product (SSCP) deviations, divided by a normalizing constant, in linear regression or canonical correlation analysis. These matrices may be used as input to the variable selection search routines anneal, genetic improve or eleaps.

Usage

```
## Default S3 method:
lmHmat(x,y,...)

## S3 method for class 'data.frame'
lmHmat(x,y,...)

## S3 method for class 'formula'
lmHmat(formula,data=NULL,...)

## S3 method for class 'lm'
lmHmat(fitdlmmodel,...)
```

Arguments

`x`	A matrix or data frame containing the variables for which the SSCP matrix is to be computed.
`y`	A matrix or data frame containing the set of fixed variables, the association of x is to be measured with.
`formula`	A formula of the form `'y ~ x1 + x2 + ...'`. That is, the response is the set of fixed variables and the right hand side specifies the variables whose subsets are to be compared.
`data`	Data frame from which variables specified in 'formula' are preferentially to be taken.
`fitdlmmodel`	An object of class `lm`, as produced by R's `lm` function.
`...`	further arguments for the method.

Details

Let x and y be two different groups of linearly independent variables observed on the same set of data units. It is well known that the association between x and y can be measured by their squared canonical correlations which may be found as the positive eigenvalues of certain matrix products. In particular, if T_x and $H_{x/y}$ denote SSCP matrices of deviations from the mean, respectively for the original x variables (T_x) and for their orthogonal projections onto the space spanned by the y's ($H_{x/y}$), then the positive eigenvalues of $T_x^{-1}H_{x/y}$ equal the squared correlations between x and y. Alternatively these correlations could also be found from $T_y^{-1}H_{y/x}$ but here, assuming a goal of comparing x's subsets for a given fixed set of y's, we will focus on the former product. `lmHmat` computes a scaled version of T_x and $H_{x/y}$ such that T_x is converted into a covariance matrix. These matrices can be used as input to the search routines `anneal`, `genetic` `improve` and `eleaps` that try to select x subsets based on several functions of their squared correlations with y. We note that when there is only one variable in the y set, this is equivalent to selecting predictors for linear regression based on the traditional coefficient of determination.

Value

A list with four items:

`mat`	The total SSCP matrix divided by nrow(x)-1
`H`	The effect SSCP matrix divided by nrow(x)-1
`r`	The expected rank of the H matrix which, under the assumption of linear independence, equals the minimum between the number of variables in the `x` and `y` sets. The true rank of `H` can be different from `r` if the linear independence condition fails.
`call`	The function call which generated the output.

See Also

anneal, genetic, improve, eleaps, lm.

Examples

```
##---------------------------------------------------------------

## 1)  An example of subset selection in the context of Multiple
## Linear Regression. Variable 5 (average price) in the Cars93 MASS
## library is to be regressed on 13 other  variables.  The goal is to
## compare subsets of these 13 variables according to their  ability
## to predict car prices.

library(MASS)
data(Cars93)
CarsHmat1 <- lmHmat(Cars93[c(7:8,12:15,17:22,25)],Cars93[5])
CarsHmat1

##$mat
##                    MPG.city  MPG.highway  EngineSize   Horsepower
##MPG.city            31.582281    28.283427   -4.1391655 -1.979799e+02
##MPG.highway         28.283427    28.427302   -3.4667602 -1.728655e+02
##EngineSize          -4.139165    -3.466760    1.0761220  3.977700e+01
##Horsepower        -197.979897  -172.865475   39.7769986  2.743079e+03
##RPM               1217.478962   997.335203 -339.1637447  1.146634e+03
##Rev.per.mile      1941.631019  1555.243104 -424.4118163 -1.561070e+04
##Fuel.tank.capacity  -14.985799   -13.743654    2.5830820  1.222536e+02
##Passengers          -2.433964    -2.583567    0.4017181  5.040907e-01
##Length             -54.673329   -42.267765   11.8197055  4.212964e+02
##Wheelbase          -25.567087   -22.375760    5.1819425  1.738928e+02
##Width              -15.302127   -12.902291    3.3992286  1.275437e+02
##Turn.circle        -12.071061   -10.202782    2.6029453  9.474252e+01
##Weight           -2795.094670 -2549.654628  517.1327139  2.282550e+04
##                         RPM Rev.per.mile Fuel.tank.capacity   Passengers
##MPG.city            1217.4790    1941.6310         -14.985799   -2.4339645
##MPG.highway          997.3352    1555.2431         -13.743654   -2.5835671
##EngineSize          -339.1637    -424.4118           2.583082    0.4017181
##Horsepower          1146.6339  -15610.7036         122.253612    0.5040907
##RPM               356088.7097  146589.3233        -652.324684 -289.6213184
##Rev.per.mile      146589.3233  246518.7295        -992.747020 -172.8003740
##Fuel.tank.capacity  -652.3247    -992.7470          10.754271    1.6085203
##Passengers          -289.6213    -172.8004           1.608520    1.0794764
```

```
##Length              -3844.9158   -5004.3139          33.063850    7.3626695
##Wheelbase           -1903.7693   -2156.2932          16.944811    4.9177186
##Width               -1217.0933   -1464.3712           9.898282    1.9237962
##Turn.circle          -972.5806   -1173.3281           7.096283    1.5037401
##Weight            -150636.1325 -215349.6757        1729.468268  339.0953717
##                       Length    Wheelbase        Width  Turn.circle
##MPG.city            -54.67333   -25.567087   -15.302127   -12.071061
##MPG.highway         -42.26777   -22.375760   -12.902291   -10.202782
##EngineSize           11.81971     5.181942     3.399229     2.602945
##Horsepower          421.29640   173.892824   127.543712    94.742520
##RPM               -3844.91585 -1903.769285 -1217.093268  -972.580645
##Rev.per.mile      -5004.31393 -2156.293245 -1464.371201 -1173.328074
##Fuel.tank.capacity   33.06385    16.944811     9.898282     7.096283
##Passengers            7.36267     4.917719     1.923796     1.503740
##Length              213.22955    82.021973    45.367929    34.780622
##Wheelbase            82.02197    46.507948    20.803062    15.899836
##Width                45.36793    20.803062    14.280739     9.962015
##Turn.circle          34.78062    15.899836     9.962015    10.389434
##Weight             6945.16129  3507.549088  1950.471599  1479.365358
##                       Weight
##MPG.city            -2795.0947
##MPG.highway         -2549.6546
##EngineSize            517.1327
##Horsepower          22825.5049
##RPM               -150636.1325
##Rev.per.mile      -215349.6757
##Fuel.tank.capacity    1729.4683
##Passengers            339.0954
##Length               6945.1613
##Wheelbase            3507.5491
##Width                1950.4716
##Turn.circle          1479.3654
##Weight             347977.8927

##$H
##                       MPG.city   MPG.highway    EngineSize   Horsepower
##MPG.city            11.1644681     9.9885440   -2.07077758  -137.938111
##MPG.highway          9.9885440     8.9364770   -1.85266802  -123.409453
##EngineSize          -2.0707776    -1.8526680    0.38408635    25.584662
##Horsepower        -137.9381108  -123.4094525   25.58466246  1704.239046
##RPM                  9.8795182     8.8389345   -1.83244599  -122.062428
##Rev.per.mile       707.3855707   632.8785101 -131.20537141 -8739.818920
##Fuel.tank.capacity  -6.7879209    -6.0729671    1.25901874    83.865437
##Passengers          -0.2008651    -0.1797085    0.03725632     2.481709
##Length             -24.5727044   -21.9845261    4.55772770   303.598201
##Wheelbase          -11.4130722   -10.2109633    2.11688849   141.009639
##Width               -5.7581866    -5.1516920    1.06802435    71.142967
##Turn.circle         -4.2281864    -3.7828426    0.78424099    52.239662
##Weight           -1275.6139645 -1141.2569026  236.59996884 15760.337110
##                         RPM Rev.per.mile Fuel.tank.capacity    Passengers
##MPG.city             9.879518    707.38557         -6.7879209  -0.200865141
##MPG.highway          8.838935    632.87851         -6.0729671  -0.179708544
##EngineSize          -1.832446   -131.20537          1.2590187   0.037256323
```

```
##Horsepower          -122.062428  -8739.81892        83.8654369   2.481708752
##RPM                    8.742457    625.97059        -6.0066801  -0.177747010
##Rev.per.mile         625.970586  44820.25860      -430.0856347 -12.726903044
##Fuel.tank.capacity    -6.006680   -430.08563         4.1270099   0.122124645
##Passengers            -0.177747    -12.72690         0.1221246   0.003613858
##Length               -21.744563  -1556.93728        14.9400378   0.442098962
##Wheelbase            -10.099510   -723.13724         6.9390706   0.205337894
##Width                 -5.095461   -364.84122         3.5009384   0.103598215
##Turn.circle           -3.741553   -267.89973         2.5707087   0.076071269
##Weight             -1128.799984 -80823.45772       775.5646486  22.950164550
##                         Length    Wheelbase        Width   Turn.circle
##MPG.city             -24.572704  -11.4130722   -5.7581866   -4.22818636
##MPG.highway          -21.984526  -10.2109633   -5.1516920   -3.78284262
##EngineSize             4.557728    2.1168885    1.0680243    0.78424099
##Horsepower           303.598201  141.0096393   71.1429669   52.23966202
##RPM                  -21.744563  -10.0995098   -5.0954608   -3.74155256
##Rev.per.mile       -1556.937281 -723.1372362 -364.8412174 -267.89973369
##Fuel.tank.capacity    14.940038    6.9390706    3.5009384    2.57070866
##Passengers             0.442099    0.2053379    0.1035982    0.07607127
##Length                54.083885   25.1198756   12.6736193    9.30612843
##Wheelbase             25.119876   11.6672121    5.8864067    4.32233724
##Width                 12.673619    5.8864067    2.9698426    2.18072961
##Turn.circle            9.306128    4.3223372    2.1807296    1.60129079
##Weight              2807.593227 1304.0186214  657.9107222  483.09812289
##                         Weight
##MPG.city            -1275.61396
##MPG.highway         -1141.25690
##EngineSize            236.59997
##Horsepower          15760.33711
##RPM                 -1128.79998
##Rev.per.mile       -80823.45772
##Fuel.tank.capacity    775.56465
##Passengers             22.95016
##Length               2807.59323
##Wheelbase            1304.01862
##Width                 657.91072
##Turn.circle           483.09812
##Weight             145747.29199

##$r
##[1] 1

##$call
##lmHmat.data.frame(x = Cars93[c(7:8, 12:15, 17:22, 25)], y = Cars93[5])


## 2)  An example of subset selection in the context of Canonical
## Correlation Analysis. Two groups of variables within the Cars93
## MASS library data set are compared. The first group (variables 4th,
## 5th and 6th) relates to price, while the second group is formed by 13
## variables that describe several technical car specifications. The
## goal is to select subsets of the second group that are optimal in
## terms of preserving the canonical correlations with the variables in
```

```
## the first group (Warning: the 3-variable "response" group is kept
## intact; subset selection is to be performed only in the 13-variable
## group).

library(MASS)
data(Cars93)
CarsHmat2 <- lmHmat(Cars93[c(7:8,12:15,17:22,25)],Cars93[4:6])

names(Cars93[4:6])
## [1] "Min.Price" "Price"     "Max.Price"

CarsHmat2

##$mat
##                      MPG.city  MPG.highway   EngineSize    Horsepower
##MPG.city             31.582281    28.283427   -4.1391655 -1.979799e+02
##MPG.highway          28.283427    28.427302   -3.4667602 -1.728655e+02
##EngineSize           -4.139165    -3.466760    1.0761220  3.977700e+01
##Horsepower         -197.979897  -172.865475   39.7769986  2.743079e+03
##RPM                1217.478962   997.335203 -339.1637447  1.146634e+03
##Rev.per.mile       1941.631019  1555.243104 -424.4118163 -1.561070e+04
##Fuel.tank.capacity   -14.985799   -13.743654    2.5830820  1.222536e+02
##Passengers           -2.433964    -2.583567    0.4017181  5.040907e-01
##Length              -54.673329   -42.267765   11.8197055  4.212964e+02
##Wheelbase           -25.567087   -22.375760    5.1819425  1.738928e+02
##Width               -15.302127   -12.902291    3.3992286  1.275437e+02
##Turn.circle         -12.071061   -10.202782    2.6029453  9.474252e+01
##Weight            -2795.094670 -2549.654628  517.1327139  2.282550e+04
##                          RPM Rev.per.mile Fuel.tank.capacity   Passengers
##MPG.city             1217.4790    1941.6310          -14.985799   -2.4339645
##MPG.highway           997.3352    1555.2431          -13.743654   -2.5835671
##EngineSize           -339.1637    -424.4118            2.583082    0.4017181
##Horsepower           1146.6339  -15610.7036          122.253612    0.5040907
##RPM                356088.7097  146589.3233         -652.324684 -289.6213184
##Rev.per.mile       146589.3233  246518.7295         -992.747020 -172.8003740
##Fuel.tank.capacity   -652.3247    -992.7470           10.754271    1.6085203
##Passengers           -289.6213    -172.8004            1.608520    1.0794764
##Length              -3844.9158   -5004.3139           33.063850    7.3626695
##Wheelbase           -1903.7693   -2156.2932           16.944811    4.9177186
##Width               -1217.0933   -1464.3712            9.898282    1.9237962
##Turn.circle          -972.5806   -1173.3281            7.096283    1.5037401
##Weight            -150636.1325 -215349.6757         1729.468268  339.0953717
##                       Length    Wheelbase        Width  Turn.circle
##MPG.city             -54.67333   -25.567087   -15.302127   -12.071061
##MPG.highway          -42.26777   -22.375760   -12.902291   -10.202782
##EngineSize            11.81971     5.181942     3.399229     2.602945
##Horsepower           421.29640   173.892824   127.543712    94.742520
##RPM                -3844.91585 -1903.769285 -1217.093268  -972.580645
##Rev.per.mile       -5004.31393 -2156.293245 -1464.371201 -1173.328074
##Fuel.tank.capacity    33.06385    16.944811     9.898282     7.096283
##Passengers             7.36267     4.917719     1.923796     1.503740
##Length               213.22955    82.021973    45.367929    34.780622
##Wheelbase             82.02197    46.507948    20.803062    15.899836
```

```
##Width                45.36793    20.803062    14.280739     9.962015
##Turn.circle          34.78062    15.899836     9.962015    10.389434
##Weight             6945.16129  3507.549088  1950.471599  1479.365358
##                       Weight
##MPG.city           -2795.0947
##MPG.highway        -2549.6546
##EngineSize           517.1327
##Horsepower         22825.5049
##RPM              -150636.1325
##Rev.per.mile     -215349.6757
##Fuel.tank.capacity  1729.4683
##Passengers           339.0954
##Length              6945.1613
##Wheelbase           3507.5491
##Width               1950.4716
##Turn.circle         1479.3654
##Weight            347977.8927

##$H
##                        MPG.city   MPG.highway    EngineSize   Horsepower
##MPG.city             12.6374638    11.1802504   -2.44856549  -149.055525
##MPG.highway          11.1802504     9.9241995   -2.15551417  -132.381671
##EngineSize           -2.4485655    -2.1555142    0.48131168    28.438641
##Horsepower         -149.0555255  -132.3816709   28.43864077  1788.168412
##RPM                 116.9463468    90.2758380  -29.90735790  -935.019669
##Rev.per.mile        850.6791690   744.7148717 -168.44221351 -9825.172173
##Fuel.tank.capacity   -7.3863845    -6.5473387    1.41367337    88.391549
##Passengers           -0.2756475    -0.2507147    0.05519028     3.036255
##Length              -29.0878749   -25.4205633    5.74148535   337.880225
##Wheelbase           -12.4579187   -11.0208656    2.38906697   148.928887
##Width                -6.8768553    -6.0641799    1.35405290    79.579106
##Turn.circle          -4.9652258    -4.3460777    0.97719452    57.833523
##Weight            -1399.0819460 -1239.6883974  268.43952022 16693.580681
##                           RPM Rev.per.mile Fuel.tank.capacity   Passengers
##MPG.city             116.946347    850.67917         -7.3863845  -0.27564745
##MPG.highway           90.275838    744.71487         -6.5473387  -0.25071469
##EngineSize           -29.907358   -168.44221          1.4136734   0.05519028
##Horsepower          -935.019669  -9825.17217         88.3915487   3.03625516
##RPM                 8930.289631  11941.01945        -51.6620352  -3.30491485
##Rev.per.mile       11941.019450  59470.19917       -490.0061258 -18.17896445
##Fuel.tank.capacity   -51.662035   -490.00613          4.3742368   0.14814085
##Passengers            -3.304915    -18.17896          0.1481409   0.01208827
##Length              -397.601848  -2033.81167         16.8646785   0.57474210
##Wheelbase            -93.828737   -830.92582          7.3783050   0.24261242
##Width                -84.771418   -472.37388          3.9523474   0.16370704
##Turn.circle          -64.578815   -345.33527          2.8839031   0.09876958
##Weight            -10423.776629 -93087.56026        826.3348263  28.56899347
##                        Length    Wheelbase        Width   Turn.circle
##MPG.city            -29.0878749  -12.4579187   -6.8768553   -4.96522585
##MPG.highway         -25.4205633  -11.0208656   -6.0641799   -4.34607767
##EngineSize            5.7414854    2.3890670    1.3540529    0.97719452
##Horsepower          337.8802249  148.9288871   79.5791065   57.83352310
##RPM                -397.6018484  -93.8287370  -84.7714184  -64.57881537
```

```
##Rev.per.mile       -2033.8116669 -830.9258201 -472.3738765 -345.33527111
##Fuel.tank.capacity    16.8646785    7.3783050    3.9523474    2.88390313
##Passengers             0.5747421    0.2426124    0.1637070    0.09876958
##Length                69.9185456   28.6482825   16.0342179   11.86931842
##Wheelbase             28.6482825   12.4615297    6.6687394    4.89477408
##Width                 16.0342179    6.6687394    3.8217667    2.73004255
##Turn.circle           11.8693184    4.8947741    2.7300425    2.01640426
##Weight              3199.4701647 1393.7884808  751.2183342  546.92139008
##                         Weight
##MPG.city            -1399.08195
##MPG.highway         -1239.68840
##EngineSize            268.43952
##Horsepower          16693.58068
##RPM                -10423.77663
##Rev.per.mile       -93087.56026
##Fuel.tank.capacity    826.33483
##Passengers             28.56899
##Length               3199.47016
##Wheelbase            1393.78848
##Width                 751.21833
##Turn.circle           546.92139
##Weight             156186.68328

##$r
##[1] 3

##$call
##lmHmat.data.frame(x = Cars93[c(7:8, 12:15, 17:22, 25)], y = Cars93[4:6])
```

`rm.coef` *Computes the RM coefficient for variable subset selection*

Description

Computes the RM coefficient, measuring the similarity of the spectral decompositions of a p-variable data matrix, and of the matrix which results from regressing all the variables on a subset of only k variables.

Usage

```
rm.coef(mat, indices)
```

Arguments

`mat`	the full data set's covariance (or correlation) matrix
`indices`	a numerical vector, matrix or 3-d array of integers giving the indices of the variables in the subset. If a matrix is specified, each row is taken to represent a different *k*-variable subset. If a 3-d array is given, it is assumed that the third dimension corresponds to different cardinalities.

Details

Computes the RM coefficient that measures the similarity of the spectral decompositions of a p-variable data matrix, and of the matrix which results from regressing those variables on a subset (given by "indices") of the variables. Input data is expected in the form of a (co)variance or correlation matrix. If a non-square matrix is given, it is assumed to be a data matrix, and its correlation matrix is used as input.

The definition of the RM coefficient is as follows:

$$RM = \sqrt{\frac{\mathrm{tr}(X^t P_v X)}{\mathrm{X^t X}}}$$

where X is the full (column-centered) data matrix and P_v is the matrix of orthogonal projections on the subspace spanned by a k-variable subset.

This definition is equivalent to:

$$RM = \sqrt{\frac{\sum_{i=1}^{p} \lambda_i (r)_i^2}{\sum_{j=1}^{p} \lambda_j}}$$

where λ_i stands for the i-th largest eigenvalue of the covariance matrix defined by X and r stands for the multiple correlation between the `i`-th Principal Component and the k-variable subset.

These definitions are also equivalent to the expression used in the code, which only requires the covariance (or correlation) matrix of the data under consideration.

The fact that `indices` can be a matrix or 3-d array allows for the computation of the RM values of subsets produced by the search functions `anneal`, `genetic` and `improve` (whose output option `$subsets` are matrices or 3-d arrays), using a different criterion (see the example below).

Value

The value of the RM coefficient.

References

Cadima, J. and Jolliffe, I.T. (2001), "Variable Selection and the Interpretation of Principal Subspaces", *Journal of Agricultural, Biological and Environmental Statistics*, Vol. 6, 62-79.

McCabe, G.P. (1986) "Prediction of Principal Components by Variable Subsets", *Technical Report 86-19, Department of Statistics, Purdue University*.

Ramsay, J.O., ten Berge, J. and Styan, G.P.H. (1984), "Matrix Correlation", *Psychometrika*, 49, 403-423.

Examples

```
## An example with a very small data set.

data(iris3)
x<-iris3[,,1]
rm.coef(var(x),c(1,3))
## [1] 0.8724422
```

```
## An example computing the RMs of three subsets produced when the
## anneal function attempted to optimize the RV criterion (using an
## absurdly small number of iterations).

data(swiss)
rvresults<-anneal(cor(swiss),2,nsol=4,niter=5,criterion="Rv")
rm.coef(cor(swiss),rvresults$subsets)

##             Card.2
##Solution 1 0.7982296
##Solution 2 0.7945390
##Solution 3 0.7649296
##Solution 4 0.7623326
```

`rv.coef` *Computes the RV-coefficient applied to the variable subset selection problem*

Description

Computes the RV coefficient, measuring the similarity (after rotations, translations and global resizing) of two configurations of n points given by: (i) observations on each of p variables, and (ii) the regression of those p observed variables on a subset of the variables.

Usage

```
rv.coef(mat, indices)
```

Arguments

`mat`	the full data set's covariance (or correlation) matrix
`indices`	a numerical vector, matrix or 3-d array of integers giving the indices of the variables in the subset. If a matrix is specified, each row is taken to represent a different *k*-variable subset. If a 3-d array is given, it is assumed that the third dimension corresponds to different cardinalities.

Details

Input data is expected in the form of a (co)variance or correlation matrix of the full data set. If a non-square matrix is given, it is assumed to be a data matrix, and its correlation matrix is used as input. The subset of variables on which the full data set will be regressed is given by `indices`.

The RV-coefficient, for a (coumn-centered) data matrix (with p variables/columns) X, and for the regression of these columns on a k-variable subset, is given by:

$$RV = \frac{\mathrm{tr}(XX^t \cdot (P_vX)(P_vX)^t)}{\sqrt{\mathrm{tr}((XX^t)^2) \cdot \mathrm{tr}(((P_vX)(P_vX)^t)^2)}}$$

where P_v is the matrix of orthogonal projections on the subspace defined by the k-variable subset.

This definition is equivalent to the expression used in the code, which only requires the covariance (or correlation) matrix of the data under consideration.

The fact that `indices` can be a matrix or 3-d array allows for the computation of the RV values of subsets produced by the search functions `anneal`, `genetic` and `improve` (whose output option `$subsets` are matrices or 3-d arrays), using a different criterion (see the example below).

Value

The value of the RV-coefficient.

References

Robert, P. and Escoufier, Y. (1976), "A Unifying tool for linear multivariate statistical methods: the RV-coefficient", *Applied Statistics*, Vol.25, No.3, p. 257-265.

Examples

```
# A simple example with a trivially small data set

data(iris3)
x<-iris3[,,1]
rv.coef(var(x),c(1,3))
## [1] 0.8659685


## An example computing the RVs of three subsets produced when the
## anneal function attempted to optimize the RM criterion (using an
## absurdly small number of iterations).

data(swiss)
rmresults<-anneal(cor(swiss),2,nsol=4,niter=5,criterion="Rm")
rv.coef(cor(swiss),rmresults$subsets)

##              Card.2
##Solution 1 0.8389669
##Solution 2 0.8663006
##Solution 3 0.8093862
##Solution 4 0.7529066
```

`tau2.coef` *Computes the Tau squared coefficient for a multivariate linear hypothesis*

Description

Computes the Tau squared index of "effect magnitude". The maximization of this criterion is equivalent to the minimization of Wilk's lambda statistic.

Usage

```
tau2.coef(mat, H, r, indices,
tolval=10*.Machine$double.eps, tolsym=1000*.Machine$double.eps)
```

Arguments

`mat`	the Variance or Total sums of squares and products matrix for the full data set.
`H`	the Effect description sums of squares and products matrix (defined in the same way as the `mat` matrix).
`r`	the Expected rank of the H matrix. See the `Details` below.
`indices`	a numerical vector, matrix or 3-d array of integers giving the indices of the variables in the subset. If a matrix is specified, each row is taken to represent a different *k*-variable subset. If a 3-d array is given, it is assumed that the third dimension corresponds to different cardinalities.
`tolval`	the tolerance level to be used in checks for ill-conditioning and positive-definiteness of the 'total' and 'effects' (H) matrices. Values smaller than `tolval` are considered equivalent to zero.
`tolsym`	the tolerance level for symmetry of the covariance/correlation/total matrix and for the effects (`H`) matrix. If corresponding matrix entries differ by more than this value, the input matrices will be considered asymmetric and execution will be aborted. If corresponding entries are different, but by less than this value, the input matrix will be replaced by its symmetric part, i.e., input matrix A becomes (A+t(A))/2.

Details

Different kinds of statistical methodologies are considered within the framework, of a multivariate linear model:

$$X = A\Psi + U$$

where X is the (nxp) data matrix of original variables, A is a known (nxp) design matrix, Ψ an (qxp) matrix of unknown parameters and U an (nxp) matrix of residual vectors. The τ^2 index is related to the traditional test statistic (Wilk's lambda statistic) and measures the contribution of each subset to an Effect characterized by the violation of a linear hypothesis of the form $C\Psi = 0$, where C is a known cofficient matrix of rank r. The Wilk's lambda statistic (λ) is given by:

$$\Lambda = \frac{det(E)}{det(T)}$$

where E is the Error matrix and T is the Total matrix. The index τ^2 is related to the Wilk's lambda statistic (Λ) by:

$$\tau^2 = 1 - \lambda^{(1/r)}$$

where r is the rank of H the Effect matrix.

The fact that `indices` can be a matrix or 3-d array allows for the computation of the τ^2 values of subsets produced by the search functions `anneal`, `genetic`, `improve` and `eleaps` (whose output option `$subsets` are matrices or 3-d arrays), using a different criterion (see the example below).

Value

The value of the τ^2 coefficient.

Examples

```
## --------------------------------------------------------------

## 1) A Linear Discriminant Analysis example with a very small data set.
## We considered the Iris data and three groups,
## defined by species (setosa, versicolor and virginica).

data(iris)
irisHmat <- ldaHmat(iris[1:4],iris$Species)
tau2.coef(irisHmat$mat,H=irisHmat$H,r=2,c(1,3))
## [1] 0.8003044

## --------------------------------------------------------------

## 2) An example computing the value of the tau_2 criterion for two
## subsets produced when the anneal function attempted to optimize
## the xi_2 criterion (using an absurdly small number of iterations).

xiresults<-anneal(irisHmat$mat,2,nsol=2,niter=2,criterion="xi2",
H=irisHmat$H,r=2)
tau2.coef(irisHmat$mat,H=irisHmat$H,r=2,xiresults$subsets)

##             Card.2
##Solution 1 0.8079476
##Solution 2 0.7907710

## --------------------------------------------------------------
```

trim.matrix

Given an ill-conditioned square matrix, deletes rows/columns until a well-conditioned submatrix is obtained.

Description

This function seeks to deal with ill-conditioned matrices, for which the search algorithms of optimal k-variable subsets could encounter numerical problems. Given a square matrix `mat` which is assumed positive semi-definite, the function checks whether it has reciprocal of the 2-norm condition number (i.e., the ratio of the smallest to the largest eigenvalue) smaller than `tolval`. If not, the matrix is considered well-conditioned and remains unchanged. If the ratio of the smallest to largest eigenvalue is smaller than `tolval`, an iterative process is begun, which deletes rows/columns (using Jolliffe's method for subset selections described on pg. 138 of the Reference below) until a principal submatrix with reciprocal of the condition number larger than `tolval` is obtained.

Usage

```
trim.matrix(mat,tolval=10*.Machine$double.eps)
```

Arguments

`mat`	a symmetric matrix, assumed positive semi-definite.
`tolval`	the tolerance value for the reciprocal condition number of matrix *mat*.

Details

For the given matrix `mat`, eigenvalues are computed. If the ratio of the smallest to the largest eigenvalue is less than `tolval`, matrix `mat` remains unchanged and the function stops. Otherwise, an iterative process is begun, in which the eigenvector associated with the smallest eigenvalue is considered and its largest (in absolute value) element is identified. The corresponding row/column are deleted from matrix `mat` and the eigendecomposition of the resulting submatrix is computed. This iterative process stops when the ratio of the smallest to largest eigenvalue is not smaller than `tolval`.

The function checks whether the input matrix is square, but not whether it is positive semi-definite. This `trim.matrix` function can be used to delete rows/columns of square matrices, until only non-negative eigenvalues appear.

Value

Output is a list with four items:

`trimmedmat`	is a principal submatrix of the original matrix, with the ratio of its smallest to largest eigenvalues no smaller than `tolval`. This matrix can be used as input for the search algorithms in this package.
`numbers.discarded`	is a list of the integer numbers of the original variables that were discarded.
`names.discarded`	is a list of the original column numbers of the variables that were discarded.
`size`	is the size of the output matrix.

Note

When the `trim.matrix` function is used to produce a well-conditioned matrix for use with the `anneal`, `genetic`, `improve` or `eleaps` functions, care must be taken in interpreting the output of those functions. In those search functions, the selected variable subsets are specified by variable numbers, and those variable numbers indicate the position of the variables in the input matrix. Hence, if a trimmed matrix is supplied to functions `anneal`, `genetic`, `improve` or `eleaps`, variable numbers refer *to the trimmed matrix*.

References

Jolliffe, I.T. (2002) *Principal Component Analysis, second edition*, Springer Series in Statistics.

Examples

```
# a trivial example, for illustration of use: creating an extra column,
# as the sum of columns in the "iris" data, and then using the function
# trim.matrix to exclude it from the data's correlation matrix

data(iris)
lindepir<-cbind(apply(iris[,-5],1,sum),iris[,-5])
colnames(lindepir)[1]<-"Sum"
cor(lindepir)

##                    Sum Sepal.Length Sepal.Width Petal.Length Petal.Width
##Sum           1.0000000    0.9409143  -0.2230928    0.9713793   0.9538850
##Sepal.Length  0.9409143    1.0000000  -0.1175698    0.8717538   0.8179411
##Sepal.Width  -0.2230928   -0.1175698   1.0000000   -0.4284401  -0.3661259
##Petal.Length  0.9713793    0.8717538  -0.4284401    1.0000000   0.9628654
##Petal.Width   0.9538850    0.8179411  -0.3661259    0.9628654   1.0000000

trim.matrix(cor(lindepir))

##$trimmedmat
##             Sepal.Length Sepal.Width Petal.Length Petal.Width
##Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
##Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
##Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
##Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
##
##$numbers.discarded
##[1] 1
##
##$names.discarded
##[1] "Sum"
##
##$size
##[1] 4

data(swiss)
lindepsw<-cbind(apply(swiss,1,sum),swiss)
colnames(lindepsw)[1]<-"Sum"
trim.matrix(cor(lindepsw))

##$lowrankmat
##                  Fertility Agriculture examination   Education   Catholic
##Fertility         1.0000000  0.35307918  -0.6458827 -0.66378886  0.4636847
##Agriculture       0.3530792  1.00000000  -0.6865422 -0.63952252  0.4010951
##Examination      -0.6458827 -0.68654221   1.0000000  0.69841530 -0.5727418
##Education        -0.6637889 -0.63952252   0.6984153  1.00000000 -0.1538589
##Catholic          0.4636847  0.40109505  -0.5727418 -0.15385892  1.0000000
##Infant.Mortality  0.4165560 -0.06085861  -0.1140216 -0.09932185  0.1754959
##                 Infant.Mortality
##Fertility             0.41655603
##Agriculture          -0.06085861
##Examination          -0.11402160
```

```
##Education             -0.09932185
##Catholic               0.17549591
##Infant.Mortality       1.00000000
##
##$numbers.discarded
##[1] 1
##
##$names.discarded
##[1] "Sum"
##
##$size
##[1] 6
```

`wald.coef` *Wald statistic for variable selection in generalized linear models*

Description

Computes the value of Wald's statistic, testing the significance of the excluded variables, in the context of variable subset selection in generalized linear models

Usage

```
wald.coef(mat, H, indices,
tolval=10*.Machine$double.eps, tolsym=1000*.Machine$double.eps)
```

Arguments

`mat`	An estimate (FI) of Fisher's information matrix for the full model variable-coefficient estimates
`H`	A matrix product of the form `FI %*% b %*% t(b) %*% FI` where `b` is a vector of variable-coefficient estimates
`indices`	a numerical vector, matrix or 3-d array of integers giving the indices of the variables in the subset. If a matrix is specified, each row is taken to represent a different *k*-variable subset. If a 3-d array is given, it is assumed that the third dimension corresponds to different cardinalities.
`tolval`	the tolerance level to be used in checks for ill-conditioning and positive-definiteness of the Fisher Information and the auxiliar (H) matrices. Values smaller than `tolval` are considered equivalent to zero.
`tolsym`	the tolerance level for symmetry of the Fisher Information and the auxiliar (H) matrices. If corresponding matrix entries differ by more than this value, the input matrices will be considered asymmetric and execution will be aborted. If corresponding entries are different, but by less than this value, the input matrix will be replaced by its symmetric part, i.e., input matrix A becomes (A+t(A))/2.

Details

Variable selection in the context of generalized linear models is typically based on the minimization of statistics that test the significance of excluded variables. In particular, the likelihood ratio, Wald's, Rao's and some adaptations of such statistics, are often proposed as comparison criteria for variable subsets of the same dimensionality. All these statistics are assympotically equivalent and can be converted into information criteria, like the AIC, that are also able to compare subsets of different dimensionalities (see references [1] and [2] for further details).

Among these criteria, Wald's statistic has some computational advantages because it can always be derived from the same (concerning the full model) maximum likelihood and Fisher information estimates. In particular, if W_{allv} is the value of the Wald statistic testing the significance of the full covariate vector, b and FI are coefficient and Fisher information estimates and H is an auxiliary rank-one matrix given by `H = FI %*% b %*% t(b) %*% FI`, it follows that the value of Wald's statistic for the excluded variables (W_{excv}) in a given subset is given by $W_{excv} = W_{allv} - tr(FI_{indices}^{-1} H_{indices})$, where $FI_{indices}$ and $H_{indices}$ are the portions of the FI and H matrices associated with the selected variables.

The FI and H matrices can be retrieved (from a glm object) by the `glmHmat` function and may be used as input to the search functions `anneal`, `genetic`, `improve` and `eleaps`. The Wald function computes the value of Wald statistc from these matrices for a subset specified by `indices`

The fact that `indices` can be a matrix or 3-d array allows for the computation of the Wald statistic values of subsets produced by the search functions `anneal`, `genetic`, `improve` and `eleaps` (whose output option `$subsets` are matrices or 3-d arrays), using a different criterion (see the example below).

Value

The value of the Wald statistic.

References

[1] Lawless, J. and Singhal, K. (1978). Efficient Screening of Nonnormal Regression Models, *Biometrics*, Vol. 34, 318-327.

[2] Lawless, J. and Singhal, K. (1987). ISMOD: An All-Subsets Regression Program for Generalized Models I. Statistical and Computational Background, *Computer Methods and Programs in Biomedicine*, Vol. 24, 117-124.

Examples

```
## --------------------------------------------------------------


##  An example of variable selection in the context of binary response
##  regression models. The logarithms and original physical measurements
##  of the "Leptograpsus variegatus crabs" considered in the MASS crabs
##  data set are used to fit a logistic model that takes the sex of each crab
##  as the response variable.

library(MASS)
data(crabs)
```

```
lFL <- log(crabs$FL)
lRW <- log(crabs$RW)
lCL <- log(crabs$CL)
lCW <- log(crabs$CW)
logrfit <- glm(sex ~ FL + RW + CL + CW  + lFL + lRW + lCL + lCW,
crabs,family=binomial)
## Warning message:
## fitted probabilities numerically 0 or 1 occurred in: glm.fit(x = X, y = Y,
## weights = weights, start = start, etastart = etastart,

lHmat <- glmHmat(logrfit)
wald.coef(lHmat$mat,lHmat$H,c(1,6,7),tolsym=1E-06)
## [1] 2.286739
## Warning message:

## The covariance/total matrix supplied was slightly asymmetric:
## symmetric entries differed by up to 6.57252030578093e-14.
## (less than the 'tolsym' parameter).
## It has been replaced by its symmetric part.
## in: validmat(mat, p, tolval, tolsym)


## ---------------------------------------------------------------

## 2) An example computing the value of the Wald statistic in a logistic
##  model for five subsets produced when a probit model was originally
##  considered

library(MASS)
data(crabs)
lFL <- log(crabs$FL)
lRW <- log(crabs$RW)
lCL <- log(crabs$CL)
lCW <- log(crabs$CW)
probfit <- glm(sex ~ FL + RW + CL + CW  + lFL + lRW + lCL + lCW,
crabs,family=binomial(link=probit))
## Warning message:
## fitted probabilities numerically 0 or 1 occurred in: glm.fit(x = X, y = Y,
## weights = weights, start = start, etastart = etastart)

pHmat <- glmHmat(probfit)
probresults <-eleaps(pHmat$mat,kmin=3,kmax=3,nsol=5,criterion="Wald",H=pHmat$H,
r=1,tolsym=1E-10)
## Warning message:

## The covariance/total matrix supplied was slightly asymmetric:
## symmetric entries differed by up to 3.14059889205964e-12.
## (less than the 'tolsym' parameter).
## It has been replaced by its symmetric part.
## in: validmat(mat, p, tolval, tolsym)

logrfit <- glm(sex ~ FL + RW + CL + CW  + lFL + lRW + lCL + lCW,
crabs,family=binomial)
```

```
## Warning message:
## fitted probabilities numerically 0 or 1 occurred in: glm.fit(x = X, y = Y,
## weights = weights, start = start, etastart = etastart)

lHmat <- glmHmat(logrfit)
wald.coef(lHmat$mat,H=lHmat$H,probresults$subsets,tolsym=1e-06)
##             Card.3
## Solution 1 2.286739
## Solution 2 2.595165
## Solution 3 2.585149
## Solution 4 2.669059
## Solution 5 2.690954
## Warning message:

## The covariance/total matrix supplied was slightly asymmetric:
## symmetric entries differed by up to 6.57252030578093e-14.
## (less than the 'tolsym' parameter).
## It has been replaced by its symmetric part.
## in: validmat(mat, p, tolval, tolsym)
```

`xi2.coef` *Computes the Xi squared coefficient for a multivariate linear hypothesis*

Description

Computes the Xi squared index of "effect magnitude". The maximization of this criterion is equivalent to the maximization of the traditional test statistic, the Bartllet-Pillai trace.

Usage

```
xi2.coef(mat, H, r, indices,
tolval=10*.Machine$double.eps, tolsym=1000*.Machine$double.eps)
```

Arguments

`mat`	the Variance or Total sums of squares and products matrix for the full data set.
`H`	the Effect description sums of squares and products matrix (defined in the same way as the `mat` matrix).
`r`	the Expected rank of the H matrix. See the `Details` below.
`indices`	a numerical vector, matrix or 3-d array of integers giving the indices of the variables in the subset. If a matrix is specified, each row is taken to represent a different *k*-variable subset. If a 3-d array is given, it is assumed that the third dimension corresponds to different cardinalities.
`tolval`	the tolerance level to be used in checks for ill-conditioning and positive-definiteness of the 'total' and 'effects' (H) matrices. Values smaller than `tolval` are considered equivalent to zero.

`tolsym` the tolerance level for symmetry of the covariance/correlation/total matrix and for the effects (H) matrix. If corresponding matrix entries differ by more than this value, the input matrices will be considered asymmetric and execution will be aborted. If corresponding entries are different, but by less than this value, the input matrix will be replaced by its symmetric part, i.e., input matrix A becomes (A+t(A))/2.

Details

Different kinds of statistical methodologies are considered within the framework, of a multivariate linear model:

$$X = A\Psi + U$$

where X is the (nxp) data matrix of original variables, A is a known (nxp) design matrix, Ψ an (qxp) matrix of unknown parameters and U an (nxp) matrix of residual vectors. The Xi squared index is related to the traditional test statistic (Bartllet-Pillai trace) and measures the contribution of each subset to an Effect characterized by the violation of a linear hypothesis of the form $C\Psi = 0$, where C is a known cofficient matrix of rank r. The Bartllet-Pillai trace (P) is given by: $P = tr(HT^{-1})$ where H is the Effect matrix and T is the Total matrix. The Xi squared index is related to Bartllet-Pillai trace (P) by:

$$\xi^2 = \frac{P}{r}$$

where r is the rank of H matrix.

The fact that `indices` can be a matrix or 3-d array allows for the computation of the Xi squared values of subsets produced by the search functions `anneal`, `genetic`, `improve` and `eleaps` (whose output option `$subsets` are matrices or 3-d arrays), using a different criterion (see the example below).

Value

The value of the ξ^2 coefficient.

Examples

```
## --------------------------------------------------------------

## 1) A Linear Discriminant Analysis example with a very small data set.
## We considered the Iris data and three groups,
## defined by species (setosa, versicolor and virginica).

data(iris)
irisHmat <- ldaHmat(iris[1:4],iris$Species)
xi2.coef(irisHmat$mat,H=irisHmat$H,r=2,c(1,3))
## [1] 0.4942503

## --------------------------------------------------------------

## 2) An example computing the value of the xi_2 criterion for two subsets
## produced when the anneal function attempted to optimize the tau_2
## criterion (using an absurdly small number of iterations).
```

```
tauresults<-anneal(irisHmat$mat,2,nsol=2,niter=2,criterion="tau2",
H=irisHmat$H,r=2)
xi2.coef(irisHmat$mat,H=irisHmat$H,r=2,tauresults$subsets)

##             Card.2
##Solution 1 0.5718811
##Solution 2 0.5232262

## --------------------------------------------------------------
```

---

zeta2.coef — *Computes the Zeta squared coefficient for a multivariate linear hypothesis*

Description

Computes the Zeta squared index of "effect magnitude". The maximization of this criterion is equivalent to the maximization of the traditional test statistic, the Lawley-Hotelling trace.

Usage

```
zeta2.coef(mat, H, r, indices,
tolval=10*.Machine$double.eps, tolsym=1000*.Machine$double.eps)
```

Arguments

`mat`	the Variance or Total sums of squares and products matrix for the full data set.
`H`	the Effect description sums of squares and products matrix (defined in the same way as the `mat` matrix).
`r`	the Expected rank of the H matrix. See the `Details` below.
`indices`	a numerical vector, matrix or 3-d array of integers giving the indices of the variables in the subset. If a matrix is specified, each row is taken to represent a different *k*-variable subset. If a 3-d array is given, it is assumed that the third dimension corresponds to different cardinalities.
`tolval`	the tolerance level to be used in checks for ill-conditioning and positive-definiteness of the 'total' and 'effects' (H) matrices. Values smaller than `tolval` are considered equivalent to zero.
`tolsym`	the tolerance level for symmetry of the covariance/correlation/total matrix and for the effects (`H`) matrix. If corresponding matrix entries differ by more than this value, the input matrices will be considered asymmetric and execution will be aborted. If corresponding entries are different, but by less than this value, the input matrix will be replaced by its symmetric part, i.e., input matrix A becomes (A+t(A))/2.

Details

Different kinds of statistical methodologies are considered within the framework, of a multivariate linear model:

$$X = A\Psi + U$$

where X is the (nxp) data matrix of original variables, A is a known (nxp) design matrix, Ψ an (qxp) matrix of unknown parameters and U an (nxp) matrix of residual vectors. The ζ^2 index is related to the traditional test statistic (Lawley-Hotelling trace) and measures the contribution of each subset to an Effect characterized by the violation of a linear hypothesis of the form $C\Psi = 0$, where C is a known cofficient matrix of rank r. The Lawley-Hotelling trace is given by: $V = tr(HE^{-1})$ where H is the Effect matrix and E is the Error matrix. The index ζ^2 is related to Lawley-Hotelling trace (V) by:

$$\zeta^2 = \frac{V}{V + r}$$

where r is the rank of H matrix.

The fact that `indices` can be a matrix or 3-d array allows for the computation of the ζ^2 values of subsets produced by the search functions `anneal`, `genetic`, `improve` and `eleaps` (whose output option `$subsets` are matrices or 3-d arrays), using a different criterion (see the example below).

Value

The value of the ζ^2 coefficient.

Examples

```
## ---------------------------------------------------------------

## 1) A Linear Discriminant Analysis example with a very small data set.
## We considered the Iris data and three groups,
## defined by species (setosa, versicolor and virginica).

data(iris)
irisHmat <- ldaHmat(iris[1:4],iris$Species)
zeta2.coef(irisHmat$mat,H=irisHmat$H,r=2,c(1,3))
## [1] 0.9211501


## ---------------------------------------------------------------

## 2) An example computing the value of the zeta_2 criterion for two
## subsets produced when the anneal function attempted to optimize
## the ccr1_2 criterion (using an absurdly small number of iterations).

ccr1results<-anneal(irisHmat$mat,2,nsol=2,niter=2,criterion="ccr12",
H=irisHmat$H,r=2)
zeta2.coef(irisHmat$mat,H=irisHmat$H,r=2,ccr1results$subsets)

##             Card.2
##Solution 1 0.9105021
##Solution 2 0.9161813
```

```
## ---------------------------------------------------------------
```

Index