Package ‘ffmanova’

January 2, 2012

Type Package

Title Fifty-fifty MANOVA

Version 0.1-1.2

Date 2007-04-20

Author Øyvind Langsrud and Bjørn-Helge Mevik

Maintainer ORPHANED

Encoding latin1

Description This package performs general linear modeling with multiple responses (MANCOVA). An overall p-value for each model term is calculated by the 50-50 MANOVA method, which handles collinear responses. Rotation testing is used to compute adjusted single response p-values according to familywise error rates and false discovery rates.

License GPL-2

URL `http://cran.r-project.org/src/contrib/Descriptions/ffmanova.html`

Repository CRAN

Date/Publication 2009-05-07 11:53:24

R topics documented:

adjust 2
dressing 3
ffmanova 4
ffmanovatest 6
fixModelMatrix 8
linregEst 9
m2c 10
manova5050 11

matlabColon 12
multiPvalues 13
multiStatistics 14
myorth 15
my_pValueF 16
norm 16
orth_D 17
print.ffmanova 18
rotationtest 18
stdize 20
unitest 21
xy_Obj 22
x_Obj 23

Index **25**

`adjust` *Adjust a predictor matrix for the presence of another matrix*

Description

`adjust` adjusts a predictor matrix X for the presence of another predictor matrix Y, by orthogonalizing X against Y.

Usage

```
adjust(X, Y)
```

Arguments

`X`	matrix. The matrix to be adjusted.
`Y`	matrix. The matrix to be adjusted for.

Details

The function can handle rank deficient matrices.

Value

A matrix with an orthogonal basis for the adjusted predictor matrix.

Author(s)

Øyvind Langsrud

dressing *Dressing data*

Description

A dataset from an experiment studying structural and rheological properties of a full fat dressing.

Usage

```
data(dressing)
```

Format

A data frame with 29 observations on the following 7 variables.

- `press` a numeric vector with values 75, 125 and 225. The homogenisation pressure.
- `stab` a numeric vector with values 0.3, 0.4 and 0.5. Amount of stabiliser.
- `emul` a numeric vector with values 0.1, 0.2 and 0.35. Amount of emulsifier.
- `day` a factor with levels `1, ..., 5`. The day the experimental run was performed on.
- `visc` a numeric vector. The measured viscosity of the dressing.
- `rheo` a matrix with 9 columns. Nine different response-parameters derived from rheological measuring. These parameters contain information about the physics of the dression (more general that viscosity).
- `pvol` a matrix with 241 columns. Particle-volume curves. Using a coulter-counter instrument fat particles were counted and their volumes were registered. These data are presented as smoothed histograms (equally spaced bins on log-scale). The total area under the curve represents the total volume of the counted fat particles. The shape of the curve reflects how the total fat volume is distributed among the different particle sizes.

Details

The data comes from an experiment in which full fat dressings were produced with different amount of stabiliser and emulsifier, and with different homogenisation pressure (se above).

A full factorial 3^3 design with two additional center points was used. The experiment was run over five days. It was unknown up front how many experimental runs could be performed each day, so the order of the runs was randomised.

For each dressing, viscosity, rheology and particle volume measurements were taken (se above).

The day is stored as a factor. The other design variables are stored as numerical variables. If one wants to use them as factors, one can use e.g. `factor(press)` in the model formula, or `dressing$press <- factor(dressing$press)` prior to calling the modelling function.

Source

The data is taken from a research project financed by a grant (131472/112) from the Norwegian Research Council. The project was managed by Stabburet, which is a major manufacturer of dressing in Norway.

ffmanova *Fifty-fifty MANOVA*

Description

General linear modeling of fixed-effects models with multiple responses is performed. The function calculates 50-50 MANOVA p-values, ordinary univariate p-values and adjusted p-values using rotation testing.

Usage

```
ffmanova(formula, data, stand = TRUE, nSim = 0, verbose = TRUE)
```

Arguments

`formula`	Model formula. See Details.
`data`	Data frame with model data.
`stand`	Logical. Standardization of responses. This option has effect on the 50-50 MANOVA testing and the calculation of `exVarSS`.
`nSim`	nonnegative integer. The number of simulations to use in the rotation tests. Can be a single nonnegative integer or a list of values for each term.
`verbose`	Logical. If `TRUE`, the rotation tests print trace information.

Details

The model is specified with `formula`, in the same way as in `lm` (except that offsets are not supported). See `lm` for details.

An overall p-value for all responses is calculated for each model term. This is done using the 50-50 MANOVA method, which is a modified variant of classical MANOVA made to handle several highly correlated responses.

Ordinary single response p-values are produced. By using rotation testing these can be adjusted for multiplicity according to familywise error rates or false discovery rates. Rotation testing is a Monte Carlo simulation framework for doing exact significance testing under multivariate normality. The number of simulation repetitions (`nSim`) must be chosen.

Unbalance is handled by a variant of Type II sums of squares, which has several nice properties:

1. Invariant to ordering of the model terms.
2. Invariant to scale changes.
3. Invariant to how the overparameterization problem of categorical variable models is solved (how constraints are defined).
4. Whether two-level factors are defined to be continuos or categorical does not influence the results.
5. Analysis of a polynomial model with a single experimental variable produce results equivalent to the results using an orthogonal polynomial.

In addition to significance testing an explained variance measure, which is based on sums of sums of squares, is computed for each model term.

Value

An object of class "ffmanova", which consists of the concatenated results from the underlying functions manova5050, rotationtests and unitests:

termNames	model term names
exVarSS	explained variances calculated from sums of squares summed over all responses
df	degrees of freedom - adjusted for other terms in model
df_om	degrees of freedom - adjusted for terms contained in actual term
nPC	number of principal components used for testing
nBU	number of principal components used as buffer components
exVarPC	variance explained by nPC components
exVarBU	variance explained by (nPC+nBU) components
pValues	50-50 MANOVA p-values
stand	logical. Whether the responses are standardised.
stat	The test statistics as t-statistics (when single degree of freedom) or F-statistics
pRaw	matrix of ordinary p-values from F- or t-testing
pAdjusted	matrix of adjusted p-values according to familywise error rates
pAdjFDR	matrix of adjusted p-values according to false discovery rates
simN	number of simulations performed for each term (same as input)

The matrices stat, pRaw, pAdjusted and pAdjFDR have one row for each model term and one column for each response.

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

References

Langsrud, Ø. (2002) 50-50 Multivariate Analysis of Variance for Collinear Responses. *The Statistician*, **51**, 305–317.

Langsrud, Ø. (2003) ANOVA for Unbalanced Data: Use Type II Instead of Type III Sums of Squares. *Statistics and Computing*, **13**, 163–167.

Langsrud, Ø. (2005) Rotation Tests. *Statistics and Computing*, **15**, 53–60.

Moen, B., Oust, A., Langsrud, Ø., Dorrell, N., Gemma, L., Marsden, G.L., Hinds, J., Kohler, A., Wren, B.W. and Rudi, K. (2005) An explorative multifactor approach for investigating global survival mechanisms of Campylobacter jejuni under environmental conditions. *Applied and Environmental Microbiology*, **71**, 2086-2094.

See also http://www.matforsk.no/ola/.

See Also

manova5050, rotationtests and unitests; the work horse functions.

Examples

```
data(dressing)

# An ANOVA model with all design variables as factors
# and with visc as the only response variable.
# Classical univariate Type II test results are produced.
ffmanova(visc ~ (factor(press) + factor(stab) + factor(emul))^2 + day,
         data = dressing)

# A second order response surface model with day as a block factor.
# The properties of the extended Type II approach is utilized.
ffmanova(visc ~ (press + stab + emul)^2 + I(press^2)+ I(stab^2)+ I(emul^2)+ day,
         data = dressing)

# 50-50 MANOVA results with the particle-volume curves as
# multivariate responses. The responses are not standardized.
ffmanova(pvol ~ (press + stab + emul)^2 + I(press^2)+ I(stab^2)+ I(emul^2)+ day,
         stand = FALSE, data = dressing)

# 50-50 MANOVA results with 9 rheological responses (standardized).
# 99 rotation simulation repetitions are performed.
res <- ffmanova(rheo ~ (press + stab + emul)^2 + I(press^2)+ I(stab^2)+ I(emul^2)+ day,
                nSim = 99, data = dressing)
res$pRaw      #  Unadjusted single responses p-values
res$pAdjusted #  Familywise error rate adjusted p-values
res$pAdjFDR   #  False discovery rate adjusted p-values

# As above, but this time 9999 rotation simulation repetitions
# are performed, but only for the model term stab^2.
res <- ffmanova(rheo ~ (press + stab + emul)^2 + I(press^2)+ I(stab^2)+ I(emul^2)+ day,
                nSim = c(0,0,0,0,0,9999,0,0,0,0,0), data = dressing)
res$pAdjusted[6,] # Familywise error rate adjusted p-values for stab^2
res$pAdjFDR[6,]   # False discovery rate adjusted p-values for stab^2

# Note that the results of the first example above can also be
# obtained by using the car package.
## Not run: Anova(lm(visc ~ (factor(press) + factor(stab) + factor(emul))^2 + day,
      data = dressing), type = "II")
## End(Not run)

# The results of the second example differ because Anova does not recognise
# linear terms (emul) as being contained in quadratic terms (I(emul^2)).
# A consequence here is that the clear significance of emul disappears.
## Not run: Anova(lm(visc ~ (press + stab + emul)^2 + I(press^2)+ I(stab^2)+ I(emul^2)+ day,
      data = dressing), type="II")
## End(Not run)
```

`ffmanovatest` *50-50 MANOVA testing*

Description

The function performs 50-50 MANOVA testing based on a matrix of hypothesis observations and a matrix of error observations.

Usage

```
ffmanovatest(modelData, errorData, stand = 0, part = c(0.9, 0.5),
             partBufDim = 0.5, minBufDim = 0, maxBufDim = 1e+08,
             minErrDf = 3, cp = -1)
```

Arguments

`modelData`	matrix of hypothesis observations
`errorData`	matrix of error observations
`stand`	Standardisation (0 or 1) of responses
`part`	The variance explained required when choosing the number of components for testing. The default value is 0.5, but to choose a single component 0.9 is required.
`partBufDim`	tuning parameter for the number of buffer components
`minBufDim`	minimum (if possible) number of buffer components
`maxBufDim`	maximum number of buffer components
`minErrDf`	minimum number of "free dimensions"
`cp`	correction parameter when "few" responses

Details

`modelData` and `errorObs` correspond to `hypObs` and `errorObs` calculated by `xy_Obj`.

Value

A list with components

`exVar1`	variance explained by `dimY` components
`exVar2`	variance explained by `dimY+bufferDim` components
`dim`	dimension of final "MANOVA-space"
`dimX`	the ordinary degrees of freedom for the test
`dimY`	number of components for testing
`bufferDim`	number of buffer components
`D`	test statistic: Wilks' Lambda
`E`	test statistic: Roy's Largest Root
`A`	test statistic: Hotelling-Lawley Trace Statistic
`M`	test statistic: Pillay-Bartlett Trace Statistic
`pD`	p-value: Wilks' Lambda
`pE`	p-value: LOWER BOUND for Roy's Largest Root
`pA`	p-value: Hotelling-Lawley Trace Statistic
`pM`	p-value: Pillay-Bartlett Trace Statistic

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

See Also

ffmanova

`fixModelMatrix` *Fix the "factor" matrix of a terms object.*

Description

The function takes the factor matrix of the terms object corresponding to a model formula and changes it so that model hierarchy is preserved also for powers of terms (e.g., `I(a^2)`).

Usage

```
fixModelMatrix(mOld)
```

Arguments

`mOld`	The factor matrix (i.e. the `"factor"` attribute) of a terms object.

Details

The ordinary model handling functions in R do not treat powers of terms (a^n) as being higher order terms (like interaction terms). `fixModelMatrix` takes the `"factor"` attribute of a terms object (usually created from a model formula) and changes it such that power terms can be treated hierarchically just like interaction terms.

The factor matrix has one row for each variable and one coloumn for each term. Originally, an entry is 0 if the term does not contain the variable. If it contains the variable, the entry is 1 if the variable should be coded with contrasts, and 2 if it should be coded with dummy variables. See `terms.object` for details.

The changes performed by `fixModelMatrix` are:

- Any 2's are changed to 1.
- In any coloumn corresponding to a term that contains `I(a^n)`, where `a` is the name of a variable and `n` is a positive integer, the element in the row corresponding to `a` is set to n. For instance, the entry of row `D` and coloumn `C:I(D^2)` is set to 2.
- Rows corresponding to `I(a^n)` are deleted.

Note that this changes the semantics of the factor matrix: `2` no longer means 'code via dummy variables'.

Value

A factor matrix.

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

See Also

terms, terms.object

Examples

```
mt <- terms(y ~ a + b + a:b + a:c + I(a^2) + I(a^3) + I(a^2):b)
print(mOld <- attr(mt, "factor"))
fixModelMatrix(mOld)
```

---

linregEst *Linear regression estimation*

Description

Function that performs multivariate multiple linear regression modelling ($Y = XB + E$) according to a principal component regression (PCR) approach where the number of components equals the number of nonzero eigenvalues (generalised inverse).

Usage

```
linregEst(X, Y)
linregStart(X, rank_lim = 1e-9)
linregEnd(Umodel, Y)
```

Arguments

X	regressor matrix
Y	response matrix
rank_lim	tuning parameter for the rank. The default value corresponds to the rank function in Matlab.
Umodel	this matrix is returned by linregStart

Details

The function linregEst performs the calculations in two steps by calling linregStart and linregEnd. The former functions function makes all calculations that can be done without knowing Y. The singular value decomposition (SVD) is an essential part of the calculations and some of the output variables are named according to SVD ('U', 'S' and 'V').

Value

`linregEst` returns a list with seven components. The first three components is returned by `linregStart` - the rest by `linregEnd`.

`Umodel`	Matrix of score values according to the PCR model.
`VmodelDivS`	Matrix that can be used to calculate `Umodel` from `X`. That is, `Umodel` equals `X %*% VmodelDivS`.
`VextraDivS1`	Matrix that can be used to check estimability. That is, predictions for a new X cannot be made if `Xnew %*% VextraDivS1` is (close to) zero.
`BetaU`	Matrix of regression parameters according to the PCR model.
`msError`	Mean square error of each response
`errorObs`	Error observations that can be used in multivariate testing
`Yhat`	Fitted values. Equals `Umodel %*% BetaU`

Note

When the number of error degrees of freedom exceeds the number of linearly independent responses, then the matrix of error observations is made so that several rows are zero. In this case the zero rows are omitted and a list with components `errorObs` and `df_error` is returned.

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

See Also

`ffmanova`

`m2c` *Conversion between matrices and partitioned matrices*

Description

Functions to convert a matrix to a list of partitioned matrices, and back again.

Usage

```
m2c(M, df = rep(1, dim(M)[2]))
c2m(CC)
c2df(CC)
```

Arguments

`M`	matrix to be partitioned according to `df`
`df`	integer vector. See Details
`CC`	list of matrices, typically the output of `m2c`

Details

`m2c` partitions a matrix into a list of matrices, by putting the first `df[1]` coloumns into the first matrix, the next `df[2]` coloumns into the second, etc.

`c2m` joins a partitioned matrix back into a single matrix. `c2m(m2c(X, df))` equals `X`.

`c2df` takes a list of matrices and returns a vector with the number of coloumns of the matrices.

Value

`m2c` returns a list of matrices.

`c2m` returns a matrix.

`c2df` returns a numeric vector.

Note

`sum(df)` must equal `ncol(X)`.

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

See Also

`ffmanova`

`manova5050` *Computation of 50-50 MANOVA results*

Description

The function takes a design-with-responses object created by `xy_Obj` and produces 50-50 MANOVA output. Results are produced for each term in the model.

Usage

```
manova5050(xyObj, stand)
```

Arguments

`xyObj`	design-with-responses object
`stand`	standardisation of responses (0 or 1)

Details

Classical multivariate ANOVA (MANOVA) are useless in many practical cases. The tests perform poorly in cases with several highly correlated responses and the method collapses when the number of responses exceeds the number of observations. 50-50 MANOVA is made to handle this problem. Principal component analysis (PCA) is an important part of this methodology. Each test is based on a separate PCA.

Value

A list with components

`termNames`	model term names
`exVarSS`	explained variances calculated from sums of squares summed over all responses
`df`	degrees of freedom - adjusted for other terms in model
`df_om`	degrees of freedom - adjusted for terms contained in actual term
`nPC`	number of principal components used for testing
`nBU`	number of principal components used as buffer components
`exVarPC`	variance explained by `nPC` components
`exVarBU`	variance explained by `(nPC+nBU)` components
`pValues`	50-50 MANOVA p-values
`stand`	logical. Whether the responses are standardised.

Note

The 50-50 MANOVA p-values are based on the Hotelling-Lawley Trace Statistic. The number of components for testing and the number of buffer components are chosen according to default rules.

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

References

Langsrud, Ø. (2002) Rotation Tests. *The Statistician*, **51**, 305–317.

See Also

`ffmanova`

`matlabColon` *Simulate Matlab's ':'*

Description

A function to simulate Matlab's ':' operator.

Usage

```
matlabColon(from, to)
```

Arguments

`from`	numeric. The start value.
`to`	numeric. The end value.

Details

`matlabCode(a,b)` returns `a:b` (R's version) unless `a > b`, in which case it returns `numeric(0)`.

Value

A numeric vector, possibly empty.

Author(s)

Bjørn-Helge Mevik

See Also

`seq`

Examples

```
identical(3:5, matlabColon(3, 5)) ## => TRUE
3:1 ## => 3 2 1
matlabColon(3, 1) ## => numeric(0)
```

`multiPvalues` *p-values from MANOVA test statistics*

Description

p-values from the four MANOVA test statistics are calculated according to the traditional F-distribution approximations (exact in some cases).

Usage

```
multiPvalues(D, E, A, M, dim, dimX, dimY)
```

Arguments

`D`	Wilks' Lambda
`E`	Roy's Largest Root
`A`	Hotelling-Lawley Trace Statistic
`M`	Pillay-Bartlett Trace Statistic
`dim`	Number of observations
`dimX`	Number of x-variables
`dimY`	Number of y-variables

Details

The parameters `dim`, `dimX` and `dimY` corresponds to a situation where the test statistics are calculated from two data matrices with zero mean (test of independence).

Value

pD	p-value: Wilks' Lambda
pE	p-value: LOWER BOUND for Roy's Largest Root
pA	p-value: Hotelling-Lawley Trace Statistic
pM	p-value: Pillay-Bartlett Trace Statistic

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

See Also

ffmanova

---

multiStatistics *MANOVA test statistics*

Description

The four classical MANOVA test statistics are calculated from a set of eigenvalues.

Usage

```
multiStatistics(ss)
```

Arguments

ss	A list of eigenvalues

Details

These eigenvalues are also known as the squared canonical correlation coefficients.

Value

A list with elements

D	Wilks' Lambda
E	Roy's Largest Root
A	Hotelling-Lawley Trace Statistic
M	Pillay-Bartlett Trace Statistic

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

`myorth` *Rank and orthonormal basis*

Description

`myorth(X)` makes an orthonormal basis for the space spanned by the columns of `X`. The number of columns returned equals `myrank(X)`, which is the rank of `X`.

Usage

```
myorth(X, tol_ = 1e-09)
myrank(X, tol_ = 1e-9)
```

Arguments

`X`	numeric matrix.
`tol_`	tuning parameter for the rank.

Details

The calculations are based on the singular value decomposition (`svd`). And `myrank(X)` is the number of singular values of `X` that are larger than `max(dim(X))*svd(x)$d[1]*tol_`.

Value

`myorth` returns a matrix, whose columns form an orthonormal basis.

`myrank` returns a single number, which is the rank of `X`.

Note

In the special case where `X` has a single column, `myorth(X)` returns `c*X` where `c` is a positive constant.

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

See Also

`svd`

my_pValueF *F-test p-values*

Description

This is simply a wrapper around `pf`: `my_pValueF(f, ny1, ny2)` is equivalent to `pf(f, ny1, ny2, lower.tail = FALSE)`.

Usage

```
my_pValueF(f, ny1, ny2)
```

Arguments

`f`	The F value
`ny1`	The numerator df's
`ny2`	The denominator df's

Value

A p-value.

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

See Also

`pf`

norm *Matrix norm.*

Description

`norm(X)` returns the largest singular value of `X`; it is equivalent to `svd(X, nu = 0, nv = 0)$d[1]`.

Usage

```
norm(X)
```

Arguments

`X`	a numeric matrix.

Value

The largest singular value of X.

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

See Also

svd

---

orth_D — *Making adjusted design matrix data*

Description

The function takes the output from modelData as input and calculates adjusted data

Usage

```
orth_D(D, model, method)
```

Arguments

D	A list containing a regressor matrix for each model term
model	The model coded as a matrix
method	Either "test" or "om"

Details

The "test" method adjusts data according to Type II* sums of squares. This is an extension of the traditional Type II method. The "om" method orthogonalises terms according to the model hierarchy. The result is a non-overparameterised representation of the model.

Value

An adjusted version of D is returned.

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

print.ffmanova *Print method for ffmanova*

Description

Print method for objects of class "ffmanova". It prints an ANOVA table.

Usage

```
## S3 method for class 'ffmanova'
print(x, digits = max(getOption("digits") - 3, 3), ...)
```

Arguments

x	"ffmanova" object. Typically created by ffmanova.
digits	positive integer. Minimum number of significant digits to be used for printing most numbers.
...	further arguments sent to the underlying printCoefmat.

Details

The function constructs an anova table, and prints it using printCoefmat with tailored arguments.

Value

Invisibly returns the original object.

Author(s)

Bjørn-Helge Mevik

See Also

ffmanova, printCoefmat

rotationtest *Rotation testing*

Description

The functions perform rotation testing based on a matrix of hypothesis observations and a matrix of error observations. Adjusted p-values according to familywise error rates and false discovery rates are calculated.

Usage

```
rotationtest(modelData, errorData, simN = 999, dfE = -1, dispsim = TRUE)
rotationtests(xyObj, nSim, verbose = TRUE)
```

Arguments

modelData	matrix of hypothesis observations
errorData	matrix of error observations
simN	Number of simulations for each test. Can be a single value or a list of values for each term.
dfE	Degrees of freedom for error needs to be specified if errorData is incomplete
dispsim	When TRUE, dots are displayed to illustrate simulation progress.
xyObj	a design-with-responses object created by xy_Obj
nSim	vector of nonnegative integers. The number of simulations to use for each term.
verbose	logical. Whether rotationtests (and rotationtest) should be verbose.

Details

modelData and errorObs correspond to hypObs and errorObs calculated by xy_Obj. These matrices are efficient representations of sums of squares and cross-products (see xy_Obj for details). This means that rotationtest can be viewed as a generalised F-test function.

rotationtests is a wrapper function that calls rotationtest for each term in the xyObj and collects the results.

Value

Both functions return a list with components

pAdjusted	adjusted p-values according to familywise error rates
pAdjFDR	adjusted p-values according to false discovery rates
simN	number of simulations performed for each term

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

References

Langsrud, Ø. (2005) Rotation Tests. *Statistics and Computing*, **15**, 53–60.

Moen, B., Oust, A., Langsrud, Ø., Dorrell, N., Gemma, L., Marsden, G.L., Hinds, J., Kohler, A., Wren, B.W. and Rudi, K. (2005) An explorative multifactor approach for investigating global survival mechanisms of Campylobacter jejuni under environmental conditions. *Applied and Environmental Microbiology*, **71**, 2086-2094.

See Also

unitest, unitests

stdize — *Centering and scaling of matrices*

Description

Function to center and/or scale the coloumns of a matrix in various ways. The coloumns can be centered with their means or with supplied values, and they can be scaled with their standard deviations or with supplied values.

Usage

```
stdize(x, center = TRUE, scale = TRUE, avoid.zero.divisor = FALSE)
```

Arguments

`x`	A matrix.
`center`	A logical, or a numeric vector. The values to subtract from each column. If `center` is `TRUE`, the mean values are used.
`scale`	A lgical, or a numeric vector. The values to divide each column with. If `scale` is `TRUE`, the standard deviations are used.
`avoid.zero.divisor`	A logical. If `TRUE`, each occurence of 0 in `scale` is replaced with a 1.

Details

`stdize` standardizes the coloumns of a matrix by subtracting their means (or the supplied values) and dividing by their standard deviations (or the supplied values).

If `avoid.zero.divisor` is `TRUE`, division-by-zero is guarded against by substituting any 0 in `center` (either calculated or supplied) with 1 prior to division.

The main difference between `stdize` and `scale` is that `stdize` divides by the standard deviations even when `center` is not `TRUE`.

Value

A matrix.

Author(s)

Bjørn-Helge Mevik and Øyvind Langsrud

See Also

`scale`

Examples

```
A <- matrix(rnorm(15, mean = 1), ncol = 3)
stopifnot(all.equal(stdize(A), scale(A), check.attributes = FALSE))

## These are different:
stdize(A, center = FALSE)
scale(A, center = FALSE)
```

`unitest` *Univariate F or t testing*

Description

The functions perform F or t testing for several responses based on a matrix of hypothesis observations and a matrix of error observations.

Usage

```
unitest(modelData, errorData, dfError = dim(errorData)[1])
unitests(xyObj)
```

Arguments

`modelData`	matrix of hypothesis observations
`errorData`	matrix of error observations
`dfError`	Degrees of freedom for error needs to be specified if `errorData` is incomplete
`xyObj`	a design-with-responses object created by `xy_Obj`

Details

`modelData` and `errorObs` correspond to `hypObs` and `errorObs` calculated by `xy_Obj`. These matrices are efficient representations of sums of squares and cross-products (see `xy_Obj` for details). This means the univariate F-statistics can be calculated straightforwardly from these input matrices. Furthermore, in the single-degree-of-freedom case, t-statistics with correct sign can be obtained.

`unitests` is a wrapper function that calls `unitest` for each term in the `xyObj` (see `xy_Obj` for details) and collects the results.

Value

`unitest` returns a list with components

`pValues`	p-values
`stat`	The test statistics as t-statistics (when single degree of freedom) or F-statistics

`unitests` returns a list with components

`pRaw`	Matrix of p-values from `unitest`, one row for each term.
`stat`	Matrix of test statistics from `unitest`, one row for each term.

Note

The function calculates the p-values by making a call to `pf`.

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

See Also

`rotationtest`, `rotationtests`

---

xy_Obj — *Creation of a design-with-responses object*

Description

The function takes an object created by `x_Obj` as input and add response values. Further initial computations for prediction and testing is made.

Usage

```
xy_Obj(xObj, Y)
```

Arguments

`xObj`	object created by `x_Obj`
`Y`	response matrix

Details

Traditionally, sums of squares and cross-products (SSC) is the multivariate generalisation of sums of squares. When there is a large number of responses this representation is inefficient and therefore linear combinations of observations (Langsrud, 2002) is stored instead, such as `errorObs`. The corresponding SSC matrix can be obtained by `t(errorObs)%*%errorObs`. When there is a large number of observations the errorObs representation is also inefficient, but it these cases it is possible to chose a representation with several zero rows. Then, errorObs is stored as a two-component list: A matrix containing the nonzero rows of errorObs and an integer representing the degrees of freedom for error (number of rows in the full errorObs matrix).

Value

A list with components

`xObj`	same as input
`Y`	same as input
`ssTotFull`	equals `sum(Y^2)`

ssTot	equals sum((center(Y))^2). That is, the total sum of squares summed over all responses.
ss	Sums of squares summed over all responses.
Beta	Output from linregEst where xObj$D_om is the regressor matrix.
Yhat	fitted values
YhatStd	standard deviations of fitted values
msError	mean square error of each response
errorObs	Error observations that can be used in multivariate testing
hypObs	Hypothesis observations that can be used in multivariate testing

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

References

Langsrud, Ø. (2002) 50-50 Multivariate Analysis of Variance for Collinear Responses. *The Statistician*, **51**, 305–317.

See Also

x_Obj.

---

x_Obj *Creation of a design matrix object*

Description

The function takes design/model information as input and performs initial computations for prediction and testing.

Usage

```
x_Obj(D, model)
```

Arguments

D	A list containing a regressor matrix for each model term
model	The model coded as a matrix

Details

See the source code of ffmanova to see how D and model are created.

Value

df_error	degrees of freedom for error
D	same as input
D_test	as D, but with Type II* adjusted model terms. Will be used for testing.
D_om	as D, but with OM-adjusted model terms. This is a non-overparameterised representation of the model. Will be used for prediction.
df_D_om	degrees of freedom according to D_om
df_D_test	degrees of freedom according to D_test
Beta_D	output from linregEst where D_om is response and where D is regressor
VmodelDivS_D	as above
VextraDivS1_D	as above
Umodel	output from linregStart where D_om is regressor
VmodelDivS	as above
VextraDivS1	as above
termNames	model term names

Author(s)

Øyvind Langsrud and Bjørn-Helge Mevik

See Also

linregEst, xy_Obj.

Index

∗Topic **algebra**
myorth, 15
norm, 16
∗Topic **array**
myorth, 15
norm, 16
stdize, 20
∗Topic **datasets**
dressing, 3
∗Topic **design**
ffmanova, 4
ffmanovatest, 6
manova5050, 11
orth_D, 17
x_Obj, 23
xy_Obj, 22
∗Topic **htest**
ffmanovatest, 6
multiPvalues, 13
multiStatistics, 14
my_pValueF, 16
rotationtest, 18
unitest, 21
∗Topic **models**
adjust, 2
ffmanova, 4
fixModelMatrix, 8
manova5050, 11
orth_D, 17
x_Obj, 23
xy_Obj, 22
∗Topic **multivariate**
ffmanova, 4
linregEst, 9
manova5050, 11
∗Topic **print**
print.ffmanova, 18
∗Topic **programming**
matlabColon, 12
∗Topic **regression**
linregEst, 9
∗Topic **utilities**
m2c, 10

adjust, 2

c2df *(*m2c*)*, 10
c2m *(*m2c*)*, 10

dressing, 3

ffmanova, 4, *8*, *10–12*, *14*, *18*
ffmanovatest, 6
fixModelMatrix, 8

linregEnd *(*linregEst*)*, 9
linregEst, 9, *24*
linregStart *(*linregEst*)*, 9
lm, *4*

m2c, 10
manova5050, *5*, 11
matlabColon, 12
multiPvalues, 13
multiStatistics, 14
my_pValueF, 16
myorth, 15
myrank *(*myorth*)*, 15

norm, 16

orth_D, 17

pf, *16*
print.ffmanova, 18
printCoefmat, *18*

rotationtest, 18, *22*
rotationtests, *5*, *22*
rotationtests *(*rotationtest*)*, 18

scale, *20*
seq, *13*
stdize, 20
svd, *15*, *17*

terms, *9*
terms.object, *8, 9*

unitest, *19*, 21
unitests, *5*, *19*
unitests *(*unitest*)*, 21

x_Obj, 23, *23*
xy_Obj, *19*, *21*, 22, *24*