Package 'miscTools'

February 14, 2012

Version 0.6-12

Date 2011/11/12

Title Miscellanneous Tools and Utilities

Author Arne Henningsen, Ott Toomet

Maintainer Arne Henningsen <arne.henningsen@googlemail.com>

Suggests Ecdat (>= 0.1-5)

Description Miscellanneous small tools and utilities

License GPL (>= 2)

URL http://www.r-project.org, http://www.micEcon.org

Repository CRAN

Date/Publication 2011-11-13 09:14:08

R topics documented:

`coefTable` *Coefficient Table*

Description

Generate Table for Coefficients, Std. Errors, t-values and P-values.

Usage

```
coefTable( coef, stdErr, df = NULL )
```

Arguments

`coef`	vector that contains the coefficients.
`stdErr`	vector that contains the standard errors of the coefficients.
`df`	degrees of freedom of the t-test used to calculate P-values.

Value

a matrix with 4 columns: coefficients, standard errors, t-values and P-values. If argument `df` is not provided, the last column (P-values) is filled with `NA`s.

Author(s)

Arne Henningsen

Examples

```
coefTable( rnorm( 10 ), 0.5 * abs( rnorm( 10 ) ), 20 )
```

colMedians *Medians of Columns*

Description

Compute the sample medians of the columns (non-rows) of a data.frame or array.

Usage

```
colMedians( x, na.rm = FALSE )
```

Arguments

x	a data.frame or array.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.

Value

A vector or array of the medians of each column (non-row) of x with dimension dim(x)[-1].

Author(s)

Arne Henningsen

See Also

rowMedians,median,colMeans.

Examples

```
data( "Electricity", package = "Ecdat" )
colMedians( Electricity )

a4 <- array( 1:120, dim = c(5,4,3,2),
   dimnames = list( c("a","b","c","d","e"), c("A","B","C","D"),
   c("x","y","z"), c("Y","Z") ) )
colMedians( a4 )
median( a4[ , "B", "x", "Z" ] )  # equal to
colMedians( a4 )[ "B", "x", "Z" ]
```

compPlot *Scatterplot to Compare two Variables*

Description

Plot a scatterplot to compare two variables.

Usage

```
compPlot( x, y, lim = NULL, ... )
```

Arguments

x	values of the first variable (on the X axis).
y	values of the second variable (on the Y axis).
lim	optional vector of two elements specifying the limits of both axes).
...	further arguments are passed to plot.

Author(s)

Arne Henningsen

Examples

```
x <- rnorm( 25 )
y <- 2 + 3 * x + rnorm( 25 )
ols <- lm( y ~ x )
compPlot( y, fitted( ols ) )
compPlot( y, fitted( ols ), lim = c( -6, 10 ) )
compPlot( y, fitted( ols ), pch = 20 )
```

ddnorm *Derivative of the Normal Distribution's Density Function*

Description

This function returns the derivative(s) of the density function of the normal (Gaussian) distribution with respect to the quantile, evaluated at the quantile(s), mean(s), and standard deviation(s) specified by arguments x, mean, and sd, respectively.

Usage

```
ddnorm( x, mean = 0, sd = 1 )
```

Arguments

x	quantile or vector of quantiles.
mean	mean or vector of means.
sd	standard deviation or vector of standard deviations.

Value

numeric value(s): derivative(s) of the density function of the normal distribution with respect to the quantile

Author(s)

Arne Henningsen

See Also

dnorm

Examples

```
ddnorm( c( -1, 0, 1 ) )
```

---

insertCol *Insert Column into a Matrix*

Description

Insert a new column into a matrix.

Usage

```
insertCol( m, c, v = NA, cName = "" )
```

Arguments

m	matrix.
c	column number where the new column should be inserted.
v	optional values of the new column.
cName	optional character string: the name of the new column.

Value

a matrix with one more column than the provided matrix m.

Author(s)

Arne Henningsen

See Also

insertRow.

Examples

```
m <- matrix( 1:4, 2 )
insertCol( m, 2, 5:6 )
```

`insertRow` *Insert Row into a Matrix*

Description

Insert a new row into a matrix.

Usage

```
insertRow( m, r, v = NA, rName = "" )
```

Arguments

`m`	matrix.
`r`	row number where the new row should be inserted.
`v`	optional values for the new row.
`rName`	optional character string: the name of the new row.

Value

a matrix with one more row than the provided matrix `m`.

Author(s)

Arne Henningsen

See Also

insertCol.

Examples

```
m <- matrix( 1:4, 2 )
insertRow( m, 2, 5:6 )
```

margEff *Method for Returning Marginal Effects*

Description

Currently, this package just defines the generic function `margEff` so that it can be used to define `margEff` methods for objects of specific classes in other packages.

Usage

```
margEff( object, ... )
```

Arguments

`object`	an object of which marginal effects should be calculated.
`...`	further arguments for methods

Author(s)

Arne Henningsen

nObs *Return number of observations for statistical models*

Description

Returns number of observations for statistical models. The default method assumes presence of a component `param$nObs` in `x`.

Usage

```
nObs(x, ...)
## Default S3 method:
nObs(x, ...)
## S3 method for class 'lm'
nObs(x, ...)
```

Arguments

`x`	a statistical model, such as created by `lm`
`...`	further arguments for methods

Details

This is a generic function. The default method returns the component `x$param$nObs`. The `lm`-method is based on qr-decomposition, in the same way as the does `summary.lm`.

Value

numeric, number of observations

Author(s)

Ott Toomet, <otoomet@econ.au.dk>

See Also

nParam

Examples

```
# Construct a simple OLS regression:
x1 <- runif(100)
x2 <- runif(100)
y <- 3 + 4*x1 + 5*x2 + rnorm(100)
m <- lm(y~x1+x2)  # estimate it
nObs(m)
```

---

nParam *Number of model parameters*

Description

This function returns the number of model parameters. The default method returns the component x$param$nParam.

Usage

```
nParam(x, free=FALSE, ...)
## Default S3 method:
nParam(x, ...)
## S3 method for class 'lm'
nParam(x, ...)
```

Arguments

x	a statistical model
free	logical, whether to report only the free parameters or the total number of parameters (default)
...	other arguments for methods

Details

Free parameters are the parameters with no equality restrictions. Some parameters may be restricted (e.g. sum of two probabilities may be restricted to equal unity). In this case the total number of parameters may depend on the normalisation.

Value

Number of parameters in the model

Author(s)

Ott Toomet, <otoomet@econ.au.dk>

See Also

nObs for number of observations

Examples

```
# Construct a simple OLS regression:
x1 <- runif(100)
x2 <- runif(100)
y <- 3 + 4*x1 + 5*x2 + rnorm(100)
m <- lm(y~x1+x2)  # estimate it
summary(m)
nParam(m) # you get 3
```

---

quasiconcavity *Test for quasiconcavity / quasiconvexity*

Description

Test wether a function is quasiconcave or quasiconvex. The bordered Hessian of this function is checked by quasiconcavity() or quasiconvexity().

Usage

```
quasiconcavity( m, tol = .Machine$double.eps )
quasiconvexity( m, tol = .Machine$double.eps )
```

Arguments

m	a bordered Hessian matrix or a list containing bordered Hessian matrices
tol	tolerance level (values between -tol and tol are considered to be zero).

Value

locigal or a logical vector (if m is a list).

Author(s)

Arne Henningsen

References

Chiang, A.C. (1984) *Fundamental Methods of Mathematical Economics*, 3rd ed., McGraw-Hill.

Examples

```
quasiconcavity( matrix( 0, 3, 3 ) )

quasiconvexity( matrix( 0, 3, 3 ) )

m <- list()
m[[1]] <- matrix( c( 0,-1,-1, -1,-2,3, -1,3,5 ), 3, 3 )
m[[2]] <- matrix( c( 0,1,-1, 1,-2,3, -1,3,5 ), 3, 3 )

quasiconcavity( m )

quasiconvexity( m )
```

---

rowMedians *Medians of Rows*

Description

Compute the sample medians of the rows of a data.frame or matrix.

Usage

```
rowMedians( x, na.rm = FALSE )
```

Arguments

x	a data.frame or matrix.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.

Value

A vector of the medians of each row of x.

Author(s)

Arne Henningsen

See Also

colMedians,median,colMeans.

Examples

```
m <- matrix( 1:12, nrow = 4 )
rowMedians( m )
```

rSquared *Calculate R squared value*

Description

Calculate R squared value.

Usage

```
rSquared( y, resid )
```

Arguments

y	vector of endogenous variables
resid	vector of residuals

Author(s)

Arne Henningsen

Examples

```
data( "Electricity", package = "Ecdat" )
reg <- lm( cost ~ q + pl + pk + pf, Electricity )
rSquared( Electricity$cost, reg$residuals )
summary( reg )$r.squared  # returns the same value
```

semidefiniteness *Positive or Negative Semidefiniteness*

Description

Check whether a symmetric matrix is positive or negative semidefinite.

Usage

```
semidefiniteness( m, positive = TRUE, tol = .Machine$double.eps,
   method = "det" )
```

Arguments

m	a quadratic matrix or a list containing quadratic matrices.
positive	logical. Check for positive (TRUE, default) or negative (FALSE) semidefiniteness.
tol	tolerance level (values between -tol and tol are considered to be zero).
method	method to test for semidefiniteness, either "det" (the textbook method: checking for the signs of the determinants of sub-matrices) or "eigen" (checking for the signs of the eigen values).

Details

Please note that a matrix can be neither positive nor negative semidefinite or positive and negative semidefinite at the same time.

Value

`semidefiniteness` returns a locigal value or a logical vector (if argument `m` is a list) indicating whether the matrix (or each of the matrices) is positive/negative (depending on argument `positive`) semidefinite.

Author(s)

Arne Henningsen

References

Chiang, A.C. (1984) *Fundamental Methods of Mathematical Economics*, 3rd ed., McGraw-Hill.

Examples

```
# a positive semidefinite matrix
semidefiniteness( matrix( 1, 3, 3 ))

# a negative semidefinite matrix
semidefiniteness( matrix(-1, 3, 3 ), positive = FALSE )

# a matrix that is positive and negative semidefinite
semidefiniteness( matrix( 0, 3, 3 ))
semidefiniteness( matrix( 0, 3, 3 ), positive = FALSE )

# a matrix that is neither positive nor negative semidefinite
semidefiniteness( matrix( 1:9, 3, 3 ))
semidefiniteness( matrix( 1:9, 3, 3 ), positive = FALSE )
```

`stdEr` *Standard deviations*

Description

Extract standard deviations from estimated models.

Usage

```
  stdEr(x, ...)
## Default S3 method:
stdEr(x, ...)
## S3 method for class 'lm'
stdEr(x, ...)
```

Arguments

x	a statistical model, such as created by lm
...	further arguments for methods

Details

stdEr is a generic function with methods for objects of "lm" class. The default method returns the square root of the diagonal of the variance-covariance matrix.

Value

numeric, the estimated standard errors of the coefficients.

Author(s)

Ott Toomet <otoomet@ut.ee>

See Also

vcov, summary.

Examples

```
data(cars)
lmRes <- lm(dist ~ speed, data=cars)
stdEr( lmRes )
```

---

sumKeepAttr *Sum of an Array While Keeping its Attributes*

Description

This function returns the sum of an numeric array (e.g. vector or matrix) while keeping its attributes.

Usage

```
sumKeepAttr( x, keepNames = FALSE, na.rm = FALSE )
```

Arguments

x	an numeric array (e.g. vector or matrix).
keepNames	logical. Should the name(s) of the element(s) ofx be assigned to the returned sum? (only relevant if codex has only one element).
na.rm	logical. Passed to sum. Should missing values be removed?

Value

the sum (see sum).

Author(s)

Arne Henningsen

See Also

sum

Examples

```
a <- 1:10
attr( a, "min" ) <- 1
attr( a, "max" ) <- 10
sum(a)
sumKeepAttr(a)
```

---

symMatrix *Symmetric Matrix*

Description

Create a Symmetric Matrix.

Usage

```
symMatrix( data = NA, nrow = NULL, byrow = FALSE,
upper = FALSE )
```

Arguments

data	an optional data vector.
nrow	the desired number of rows and columns.
byrow	logical. If 'FALSE' (the default) the matrix is filled by columns, otherwise the matrix is filled by rows.
upper	logical. If 'FALSE' (the default) the lower triangular part of the matrix (including the diagonal) is filled, otherwise the upper triangular part of the matrix is filled.

Value

a symmetric matrix.

Author(s)

Arne Henningsen

See Also

matrix, lower.tri.

Examples

```
# fill the lower triangular part by columns
symMatrix( 1:10, 4 )
# fill the upper triangular part by columns
symMatrix( 1:10, 4, upper = TRUE )
# fill the lower triangular part by rows
symMatrix( 1:10, 4, byrow = FALSE )
```

`triang` *Upper triangular matrix from a vector*

Description

Creates an upper triangular square matrix from a vector.

Usage

```
triang( v, n )
```

Arguments

v	vector
n	desired dimension of the returned square matrix

Note

If the vector has less elements than the upper triangular matrix, the last elements are set to zero.

Author(s)

Arne Henningsen

See Also

`veclipos`.

Examples

```
v <- c( 1:5 )
triang( v, 3 )
```

vecli *Vector of linear independent values*

Description

Returns a vector containing the linear independent elements of a symmetric matrix (of full rank).

Usage

```
vecli( m )
```

Arguments

m	symmetric matrix

Author(s)

Arne Henningsen

See Also

veclipos.

Examples

```
# a symmetric n x n matrix
m <- cbind(c(11,12,13),c(12,22,23),c(13,23,33))
vecli(m)  # returns: 11 12 13 22 23 33
```

vecli2m *Convert vector of linear independent values into a Matrix*

Description

Converts a vector into a symmetric matrix that the original vector contains the linear independent values of the returned symmetric matrix.

Usage

```
vecli2m( v )
```

Arguments

v	a vector.

Author(s)

Arne Henningsen

See Also

vecli, veclipos.

Examples

```
v <- c( 11, 12, 13, 22, 23, 33 )
vecli2m( v )
```

---

veclipos *Position in a vector of linear independent values*

Description

Returns the position of the [i,j]th element of a symmetric n × n matrix that this element has in a vector of the linear independent values of the matrix.

Usage

```
veclipos( i, j, n )
```

Arguments

i	row of the element in the matrix.
j	column of the element in the matrix.
n	dimension of the matrix.

Note

A symmetric n × n matrix has n*(n+1)/2 independent values.
The function is: n*(n-1)/2-((n-min(i,j))*(n-min(i,j)+1)/2)+max(i,j)

Author(s)

Arne Henningsen

See Also

vecli, vecli2m.

Examples

```
veclipos( 1, 2, 3 )  # returns: 2
```

Index