Package 'fastmatrix'

August 10, 2020

Type Package

Title Fast Computation of some Matrices Useful in Statistics

Version 0.2

Description Small set of functions to fast computation of some matrices and operations useful in statistics.

Author Felipe Osorio [aut, cre] (<https://orcid.org/0000-0002-4675-5201>), Alonso Ogueda [aut]

Maintainer Felipe Osorio <felipe.osorios@usm.cl>

Depends R (>= 2.10)

License GPL-3

URL https://faosorios.github.io/fastmatrix/

LazyLoad yes

NeedsCompilation yes

R topics documented:

`dupl.cross` *Matrix crossproduct envolving the duplication matrix*

Description

Given the order of two duplication matrices and matrix `x`, this function performs the operation: `y <-t(Dn) %*% x %*% Dk`, where `Dn` and `Dk` are duplication matrices of order n and k, respectively.

Usage

```
dupl.cross(n = 1, k = n, x = NULL)
```

Arguments

n	order of the duplication matrix used pre-multiplying x.
k	order of the duplication matrix used post-multiplying x. By default k = n is used.
x	numeric matrix, this argument is required.

Details

This function calls dupl.prod to performs the matrix multiplications required but without forming any duplication matrices.

See Also

dupl.prod

Examples

```
D2 <- duplication(n = 2, matrix = TRUE)
D3 <- duplication(n = 3, matrix = TRUE)
x <- matrix(1, nrow = 9, ncol = 4)
y <- t(D3) %*% x %*% D2

z <- dupl.cross(n = 3, k = 2, x) # D2 and D3 are not stored
all(z == y) # matrices y and z are equal!

x <- matrix(1, nrow = 9, ncol = 9)
z <- dupl.cross(n = 3, x = x) # same matrix is used to pre- and post-multiplying x
z # print result
```

---

dupl.info *Compact information to construct the duplication matrix*

Description

This function provides the minimum information required to create the duplication matrix.

Usage

```
dupl.info(n = 1, condensed = TRUE)
```

Arguments

n	order of the duplication matrix.
condensed	logical. Information should be returned in compact form?

Details

This function returns a list containing two vectors that represent an element of the duplication matrix and is accesed by the indexes in vectors `row` and `col`. This information is used by function `dupl.prod` to do some operations involving the duplication matrix without forming it. This information also can be obtained using function `duplication`

Value

A list containing the following elements:

`row`	vector of indexes, each entry represents the row index of the duplication matrix. Only present if `condensed = FALSE`.
`col`	vector of indexes, each entry represents the column index of the duplication matrix.
`order`	order of the duplication matrix.

See Also

`duplication` `dupl.prod`

Examples

```
z <- dupl.info(n = 3, condensed = FALSE)
z # where are the ones in duplication of order 3?

D3 <- duplication(n = 3, matrix = TRUE)
D3 # only recommended if n is very small
```

---

dupl.prod *Matrix multiplication envolving the duplication matrix*

Description

Given the order of a duplication and matrix `x`, performs one of the matrix-matrix operations:

- `y <-D %*% x`, or
- `y <-t(D) %*% x`, or
- `y <-x %*% D`, or
- `y <-x %*% t(D)`,

where `D <-duplication(n,matrix = TRUE)` is the duplication matrix of order n. The main aim of `dupl.prod` is to do this matrix multiplication **without forming** the duplication matrix.

Usage

```
dupl.prod(n = 1, x, transposed = FALSE, side = "left")
```

Arguments

n	order of the duplication matrix.
x	numeric matrix (or vector).
transposed	logical. Duplication matrix should be transposed?
side	a string selecting if duplication matrix is pre-multiplying x, that is side = "left" or post-multiplying x, by using side = "right".

Details

Underlying C code only uses information provided by dupl.info to performs the matrix multiplication. The duplication matrix is **never** created.

See Also

duplication

Examples

```
D4 <- duplication(n = 4, matrix = TRUE)
x <- matrix(1, nrow = 16, ncol = 2)
y <- crossprod(D4, x)

z <- dupl.prod(n = 4, x, transposed = TRUE) # D4 is not stored
all(z == y) # matrices y and z are equal!
```

---

duplication *Duplication matrix*

Description

This function returns the duplication matrix of order n which transforms, for a symmetric matrix x, vech(x) into vec(x).

Usage

```
duplication(n = 1, matrix = FALSE, condensed = FALSE)
```

Arguments

n	order of the duplication matrix.
matrix	a logical indicating whether the duplication matrix will be returned.
condensed	logical. Information should be returned in compact form?.

Details

This function is a wrapper function for the function dupl.info. This function provides the minimum information required to create the duplication matrix. If option matrix = FALSE the duplication matrix is stored in two vectors containing the coordinate list of indexes for rows and columns. Option condensed = TRUE only returns vector of indexes for the columns of duplication matrix.

Warning: matrix = TRUE is **not** recommended, unless the order n be small. This matrix can require a huge amount of storage.

Value

Returns an n^2 by $n(n+1)/2$ matrix.

References

Magnus, J.R., and Neudecker, H. (1980). The elimination matrix, some lemmas and applications. *SIAM Journal on Algebraic Discrete Methods* **1**, 422-449.

Magnus, J.R., and Neudecker, H. (2007). *Matrix Differential Calculus with Applications in Statistics and Econometrics*, 3rd Edition. Wiley, New York.

See Also

`dupl.info`

Examples

```
z <- duplication(n = 100, condensed = TRUE)
object.size(z) # 40.5 Kb of storage

z <- duplication(n = 100, condensed = FALSE)
object.size(z) # 80.6 Kb of storage

D100 <- duplication(n = 100, matrix = TRUE)
object.size(D100) # 202 Mb of storage, do not request this matrix!

D3 <- duplication(3, matrix = TRUE)
a <- matrix(c( 1, 2, 3,
               2, 3, 4,
               3, 4, 5), nrow = 3)
upper <- a[upper.tri(a, diag = TRUE)]
v <- D3 %*% upper
as.vector(v)
```

`hadamard` *Hadamard product of two matrices*

Description

This function returns the Hadamard or element-wise product of two matrices `x` and `y`, that have the same dimensions.

Usage

```
hadamard(x, y)
```

Arguments

`x`	a numeric matrix or vector.
`y`	a numeric matrix or vector.

Value

A matrix with the same dimension of x (and y) which corresponds to the element-by-element product of the two matrices.

References

Styan, G.P.H. (1973). Hadamard products and multivariate statistical analysis, *Linear Algebra and Its Applications* **6**, 217-240.

Examples

```
x <- matrix(rep(1:10, times = 5), ncol = 5)
y <- matrix(rep(1:5, each = 10), ncol = 5)
z <- hadamard(x, y)
z
```

---

sweep.operator *Gauss-Jordan sweep operator for symmetric matrices*

Description

Perform the sweep operation (or reverse sweep) on the k-th element of the diagonal of a symmetric matrix.

Usage

```
sweep.operator(x, k = 1, reverse = FALSE)
```

Arguments

x	a symmetric matrix.
k	element of the diagonal which will be sweeped.
reverse	logical. If reverse = TRUE the reverse sweep is performed.

Details

The symmetric sweep operator is a powerful tool in computational statistics with uses in stepwise regression, conditional multivariate normal distributions, MANOVA, and more.

Value

a square matrix of the same order as x.

References

Goodnight, J.H. (1979). A tutorial on the SWEEP operator. *The American Statistician* **33**, 149-158.

Examples

```
x <- matrix(rnorm(1000 * 100), ncol = 100)
xx <- crossprod(x)
y <- sweep.operator(xx, k = 1)
```

vec *vectorization of a matrix*

Description

This function returns a vector obtained by stacking the columns of x

Usage

```
vec(x)
```

Arguments

x	a numeric matrix.

Value

Let x be a n by m matrix, then `vec(x)` is a nm-dimensional vector.

Examples

```
x <- matrix(rep(1:10, each = 10), ncol = 10)
x
y <- vec(x)
y
```

vech *vectorization the lower triangular part of a square matrix*

Description

This function returns a vector obtained by stacking the lower triangular part of a square matrix.

Usage

```
vech(x)
```

Arguments

x	a square matrix.

Value

Let x be a n by n matrix, then `vech(x)` is a $n(n+1)/2$-dimensional vector.

Examples

```
x <- matrix(rep(1:10, each = 10), ncol = 10)
x
y <- vech(x)
y
```

Index