Package ‘MultiStatM’

July 26, 2024

Type Package

Title Multivariate Statistical Methods

Version 2.0.0

Imports arrangements, Matrix, MASS, stats, mvtnorm

Description Algorithms to build set partitions and commutator matrices and their use in the construction of multivariate d-
Hermite polynomials; estimation and derivation of theoretical vector moments and vector cumulants of multivariate distributions; conversion formulae for multivariate moments and cumulants. Applications to
estimation and derivation of multivariate measures of skewness and kurtosis; estimation and derivation of asymptotic
covariances for d-variate Hermite polynomials, multivariate moments and cumulants and measures of skewness and kurtosis.
The formulae implemented are discussed in Terdik (2021, ISBN:9783030813925), ``Multivariate Statistical Methods''.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 2.10)

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation no

Author Gyorgy Terdik [aut],
Emanuele Taufer [aut, cre]

Maintainer Emanuele Taufer <emanuele.taufer@unitn.it>

Repository CRAN

Date/Publication 2024-07-26 15:00:02 UTC

Contents

`CommutatorIndx` *Commutator Index*

Description

This function calculates the commutator index based on the specified type. The available types are "Kmn", "Kperm", "Mixing", and "Moment". Depending on the selected type, the corresponding specific function is called.

Usage

```
CommutatorIndx(Type, ...)
```

Arguments

`Type`	a string specifying the type of commutator index to be calculated. Must be one of "Kmn", "Kperm", "Mixing", or "Moment".
`...`	additional arguments passed to the specific commutator function.

Details

The function ‘CommutatorIndx‘ acts as a wrapper to call specific commutator functions based on the input ‘Type‘.

Type "Kmn":

Parameters:
- `m` - Row-dimension.
- `n` - Col-dimension.

Return: A vector of indexes to provide the commutation, transforming vec A to vec of the transposed A.

Type "Kperm":

Parameters:
- `perm` - Vector indicating the permutation of the order in the Kronecker product.
- `dims` - Vector indicating the dimensions of the vectors.

Return: An index vector to produce the permutation of the Kronecker products of vectors of any length.

Type "Mixing":

Parameters:
- `x` - A vector of dimension `prod(d1)*prod(d2)`.
- `d1` - Dimension of the first group of vectors.
- `d2` - Dimension of the second group of vectors.

Return: A vector Kx representing the product of the moment commutator and the vector x.

Type "Moment":

Parameters:
- `x` - A vector of length `d^n` where n is the length of `el_rm`.
- `el_rm` - Type of a partition.
- `d` - Dimensionality of the underlying multivariate distribution.

Return: A vector Kx representing the product of the moment commutator and the vector x.

Value

A vector representing the commutator index.

See Also

Other Commutators: `CommutatorMatr()`

Examples

```
# Kmn example
A <- 1:6
CommutatorIndx(Type = "Kmn", m = 3, n = 2)

# Kperm example
a1 <- c(1, 2)
a2 <- c(2, 3, 4)
a3 <- c(1, 3)
p1 <- a1 %x% a2 %x% a3
CommutatorIndx(Type = "Kperm", perm = c(3, 1, 2), dims = c(2, 3, 2))

# Mixing example
d1 <- c(2, 3, 2)
d2 <- c(3, 2, 2)
x <- 1:(prod(d1) * prod(d2))
CommutatorIndx(Type = "Mixing", x = x, d1 = d1, d2 = d2)

# Moment example
n <- 4
r <- 2
m <- 1
d <- 2
PTA <- PartitionTypeAll(n)
el_r <- PTA$eL_r[[r]][m, ]
x <- 1:d^n
CommutatorIndx(Type = "Moment", x = x, el_rm = el_r, d = d)
```

`CommutatorMatr` *Commutator Matrix*

Description

This function generates various types of commutator matrices.

Usage

```
CommutatorMatr(Type, ...)
```

Arguments

`Type`	A string specifying the type of commutator matrix. Choices are "Kmn", "Kperm", "Mixing", or "Moment".
`...`	Additional arguments specific to the type of commutator matrix (see Details).

Details

The function `CommutatorMatr` supports the following types of commutator matrices:

Kmn **Description:** Transforms `vec(A)` to `vec(A^T)`, where `A^T` is the transpose of matrix `A`. An option for sparse matrix is provided. By default, a non-sparse matrix is produced. Using sparse matrices increases computation times but requires far less memory. **Arguments:**

- `m` **(integer)** Number of rows of the first matrix.
- `n` **(integer)** Number of columns of the first matrix.
- `useSparse` **(logical, optional)** If TRUE, returns a sparse matrix. Default is FALSE.

Kperm **Description:** Generates a commutation matrix for a specified permutation of matrix dimensions. An option for sparse matrix is provided. By default, a non-sparse matrix is produced. Using sparse matrices increases computation times but requires far less memory. **Arguments:**

- `perm` **(integer vector)** The permutation vector.
- `dims` **(integer vector)** The dimensions of the matrices involved.
- `useSparse` **(logical, optional)** If TRUE, returns a sparse matrix. Default is FALSE.

Mixing **Description:** Generates the Mixing commutation matrix used in linear algebra transformations involving tensor products. An option for sparse matrix is provided. By default, a non-sparse matrix is produced. Using sparse matrices increases computation times but requires far less memory. **Arguments:**

- `d1` **(integer vector)** Dimensions of the first set.
- `d2` **(integer vector)** Dimensions of the second set.
- `useSparse` **(logical, optional)** If TRUE, returns a sparse matrix. Default is FALSE.

Moment **Description:** Generates the Moment commutation matrix based on partitioning of moments. An option for sparse matrix is provided. By default, a non-sparse matrix is produced. Using sparse matrices increases computation times but requires far less memory. **Arguments:**

- `el_rm` **(integer vector)** Elements of the partition.
- `d` **(integer)** Dimension of the partition.
- `useSparse` **(logical, optional)** If TRUE, returns a sparse matrix. Default is FALSE.

Value

Depending on the type:

Kmn A commutation matrix of dimension $mn \times mn$. If `useSparse=TRUE`, an object of class "dgCMatrix" is produced.

Kperm A square permutation matrix of size `prod(dims)`. If `useSparse=TRUE`, an object of class "dgCMatrix" is produced.

Mixing A square matrix of dimension `prod(d1) * prod(d2)`. If `useSparse=TRUE`, an object of class "dgCMatrix" is produced.

Moment A commutator matrix for moment formulae.

See Also

Other Commutators: `CommutatorIndx()`

Examples

```
# Example for Kmn
CommutatorMatr("Kmn", m = 3, n = 2)

# Example for Kperm
dims <- c(2, 3, 2)
perm <- c(1, 3, 2)
CommutatorMatr("Kperm", perm = perm, dims = dims)

# Example for Mixing
d1 <- c(2, 3, 2)
d2 <- c(3, 2, 2)
CommutatorMatr("Mixing", d1 = d1, d2 = d2)

# Example for Moment
n <- 4
r <- 2
m <- 1
d <- 2
PTA <- PartitionTypeAll(n)
el_r <- PTA$eL_r[[r]][m,]
CommutatorMatr("Moment", el_r = el_r, d = d)
```

Cum2Mom *Convert cumulants to moments (univariate and multivariate)*

Description

Obtains a vector of moments from a vector of cumulants for either univariate or multivariate data.

Usage

```
Cum2Mom(cumulants, Type = c("Univariate", "Multivariate"))
```

Arguments

cumulants	Either a vector of univariate cumulants or a list of vectors of multivariate cumulants.
Type	A character string specifying the type of cumulants provided. Use "Univariate" for univariate cumulants and "Multivariate" for multivariate cumulants.

Value

The vector of moments if Type is "Univariate" or the list of vectors of moments if Type is "Multivariate".

References

Gy. Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021. Section 3.4.

See Also

Other Moments and cumulants: EVSKSkewNorm(), EVSKUniS(), Mom2Cum(), MomCumCFUSN(), MomCumSkewNorm(), MomCumUniS(), MomCumZabs()

Examples

```
# Univariate example
cum_x <- c(1, 2, 3, 4)
Cum2Mom(cum_x, Type = "Univariate")

# Multivariate example
cum <- list(c(0,0), c(1,0,0,1), c(rep(0,8)), c(rep(0,16)), c(rep(0,32)))
Cum2Mom(cum, Type = "Multivariate")
```

EliminIndx *Distinct values selection vector*

Description

Eliminates the duplicated/q-plicated elements in a T-vector of multivariate moments and cumulants. Produces the same results as EliminMatr. Note EliminIndx does not provide the same results as unique()

Usage

```
EliminIndx(d, q)
```

Arguments

d	dimension of a vector x
q	power of the Kronecker product

Value

A vector of indexes of the distinct elements in the T-vector

References

Gy. Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021. Section 1.3.2 Multi-Indexing, Elimination, and Duplication, p.21,(1.32)

See Also

Other Matrices and commutators: EliminMatr(), QplicIndx(), QplicMatr(), SymIndx(), SymMatr(), UnivMomCum()

Examples

```
x<-c(1,0,3)
y<-kronecker(x,kronecker(x,x))
y[EliminIndx(3,3)]
## Not the same results as
unique(y)
```

EliminMatr *Elimination Matrix*

Description

Eliminates the duplicated/q-plicated elements in a T-vector of multivariate moments and cumulants.

Usage

```
EliminMatr(d, q, useSparse = FALSE)
```

Arguments

d	dimension of a vector x
q	power of the Kronecker product
useSparse	TRUE or FALSE.

Value

Elimination matrix of order $\eta_{d,q} \times d^q = \binom{d+q-1}{q}$. If `useSparse=TRUE` an object of the class "dgCMatrix" is produced.

References

Gy. Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021. Section 1.3.2 Multi-Indexing, Elimination, and Duplication, p.21,(1.32)

See Also

Other Matrices and commutators: `EliminIndx()`, `QplicIndx()`, `QplicMatr()`, `SymIndx()`, `SymMatr()`, `UnivMomCum()`

Examples

```
x<-c(1,2,3)
y<-kronecker(kronecker(x,x),x)
## Distinct elements of y
z<-as.matrix(EliminMatr(3,3))%*%y
## Restore eliminated elements in z
as.vector(QplicMatr(3,3)%*%z)
```

`EVSKSkewNorm` *EVSK multivariate Skew Normal*

Description

Computes the theoretical values of the mean vector, covariance, skewness vector, total skenwness, kurtosis vector and total kurtosis for the multivariate Skew Normal distribution

Usage

```
EVSKSkewNorm(omega, alpha)
```

Arguments

`omega`	A $d \times d$ correlation matrix
`alpha`	shape parameter d-vector

Value

A list of theoretical values for the mean vector, covariance, skewness vector, total skenwness, kurtosis vector and total kurtosis

References

Gy.Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021 (5.5) p.247

S. R. Jammalamadaka, E. Taufer, Gy. Terdik. On multivariate skewness and kurtosis. Sankhya A, 83(2), 607-644.

See Also

Other Moments and cumulants: `Cum2Mom()`, `EVSKUniS()`, `Mom2Cum()`, `MomCumCFUSN()`, `MomCumSkewNorm()`, `MomCumUniS()`, `MomCumZabs()`

Examples

```
alpha<-c(10,5,0)
omega<-diag(3)
EVSKSkewNorm(omega,alpha)
```

EVSKUniS — *EVSK of the Uniform distribution on the sphere or its modulus*

Description

Cumulants (up to the 4th order), skewness, and kurtosis of the d-variate Uniform distribution on the sphere or the modulus of the d-variate Uniform distribution on the sphere.

Usage

```
EVSKUniS(d, nCum = TRUE, Type = c("Standard", "Modulus"))
```

Arguments

`d`	dimensions
`nCum`	if it is FALSE then moments (up to the 4th order) are calculated.
`Type`	specify the type of distribution: "Standard" for the Uniform distribution on the sphere, or "Modulus" for the modulus of the Uniform distribution on the sphere.

Value

A list of computed moments and cumulants.

When Type is "Standard":

`EU1`	Mean vector
`varU`	Covariance matrix
`Skew.U`	Skewness vector (always zero)
`Skew.tot`	Total skewness (always zero)
`Kurt.U`	Kurtosis vector
`Kurt.tot`	Total kurtosis

When Type is "Modulus":

`EU1`	Mean vector
`varU`	Covariance matrix
`EU.k`	List of moments up to 4th order
`cumU.k`	List of cumulants up to 4th order
`skew.U`	Skewness vector
`kurt.U`	Kurtosis vector

References

Gy. Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021 Proposition 5.3 p.297

S. R. Jammalamadaka, E. Taufer, Gy. Terdik. On multivariate skewness and kurtosis. Sankhya A, 83(2), 607-644.

Gy. Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021, Lemma 5.12 p.298

See Also

Other Moments and cumulants: `Cum2Mom()`, `EVSKSkewNorm()`, `Mom2Cum()`, `MomCumCFUSN()`, `MomCumSkewNorm()`, `MomCumUniS()`, `MomCumZabs()`

Examples

```
# Example for Standard type
EVSKUniS(d=3, Type="Standard")

# Example for Modulus type
EVSKUniS(d=3, Type="Modulus")
```

HermiteCoeff *Coefficients of Hermite polynomials*

Description

Provides the coefficients of Hermite polynomials, either univariate or multivariate.

Usage

```
HermiteCoeff(Type, N, d = NULL)
```

Arguments

Type	A character string specifying the type of Hermite polynomial. Must be either "Univariate" or "Multivariate".
N	The order of polynomial. Required for both types.
d	The dimension of the d-variate X. Required only for multivariate type.

Value

For 'Type = "Univariate"', returns a vector of coefficients of x^N, x^{N-2}, etc. For 'Type = "Multivariate"', returns a list of matrices of coefficients for the d-variate polynomials from 1 to N.

References

Gy. Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021. Sections 4.4 (4.24) and 4.6.2, p. 223, Remark 4.8

See Also

Other Hermite Polynomials: HermiteCov12(), HermiteN(), HermiteN2X()

Examples

```
# Univariate example
H_uni <- HermiteCoeff(Type = "Univariate", N = 5)

# Multivariate example
N <- 5; d <- 3
H_multi <- HermiteCoeff(Type = "Multivariate", N = N, d = d)
X <- c(1:3)
X3 <- kronecker(X, kronecker(X, X))
X5 <- kronecker(X3, kronecker(X, X))
Idv <- as.vector(diag(d)) # vector of variance matrix
# value of H5 at X is
vH5 <- H_multi[[1]] %*% X5 + H_multi[[2]] %*% kronecker(Idv, X3) +
  H_multi[[3]] %*% kronecker(kronecker(Idv, Idv), X)
```

HermiteCov12 — *Covariance matrix for multivariate T-Hermite polynomials*

Description

Computation of the covariance matrix between d-variate T-Hermite polynomials $H_N(X_1)$ and $H_N(X_2)$.

Usage

```
HermiteCov12(SigX12, N)
```

Arguments

SigX12	Covariance matrix of the Gaussian vectors X1 and X2 respectively of dimensions d1 and d2
N	Common degree of the multivariate Hermite polynomials

Value

Covariance matrix of $H_N(X_1)$ and $H_N(X_2)$

References

Gy.Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021. (4.59), (4.66),

See Also

Other Hermite Polynomials: HermiteCoeff(), HermiteN(), HermiteN2X()

Examples

```
Covmat<-matrix(c(1,0.8,0.8,1),2,2)
Cov_X1_X2 <- HermiteCov12(Covmat,3)
```

`HermiteN` *Hermite Polynomials (Univariate and Multivariate)*

Description

Computes either univariate or multivariate Hermite polynomials up to a specified order.

Usage

```
HermiteN(x, N, Type, sigma2 = 1, Sig2 = diag(length(x)))
```

Arguments

`x`	A scalar (for univariate) or a vector (for multivariate) at which to evaluate the Hermite polynomials.
`N`	The maximum order of the polynomials.
`Type`	A character string specifying the type of Hermite polynomials to compute. Can be either "Univariate" or "Multivariate".
`sigma2`	The variance for univariate Hermite polynomials. Default is 1. (Only used if Type is "Univariate").
`Sig2`	The covariance matrix for multivariate Hermite polynomials. Default is the unit matrix diag(length(x)). (Only used if Type is "Multivariate").

Details

Depending on the value of the ‘Type‘ parameter, this function computes either the univariate or the multivariate Hermite polynomials.

Value

Depending on the type, the function returns:

- `Univariate`: A vector of univariate Hermite polynomials with degrees from 1 to N evaluated at x.
- `Multivariate`: A list of multivariate polynomials of order from 1 to N evaluated at vector x.

References

Gy.Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021. Section 4.1 (for univariate), Section 4.6.2, (4.73), p.223 (for multivariate).

See Also

Other Hermite Polynomials: HermiteCoeff(), HermiteCov12(), HermiteN2X()

Examples

```
# Univariate example
HermiteN(x = 1, N = 3, Type = "Univariate")

# Multivariate example
HermiteN(x = c(1, 3), N = 3, Type = "Multivariate", Sig2 = diag(2))
```

HermiteN2X *Inverse Hermite Polynomial*

Description

Compute the inverse of univariate or multivariate Hermite polynomials.

Usage

```
HermiteN2X(Type, H_N, N, Sig2 = NULL)
```

Arguments

Type	A string specifying the type of Hermite polynomial inversion. Must be either "Univariate" or "Multivariate".
H_N	Input Hermite polynomials. For univariate, it is a vector. For multivariate, it is a list.
N	The highest polynomial order.
Sig2	The variance matrix of x for multivariate, or variance for univariate. Defaults to identity matrix for multivariate and 1 for univariate.

Details

This function computes the powers of x when Hermite polynomials are given. Depending on the type specified, it handles either univariate or multivariate Hermite polynomials.

Value

A list of x powers: x, $x^{\otimes 2}$, ... , $x^{\otimes N}$ for multivariate, or a vector of x powers: x^n, $n = 1 : N$ for univariate.

References

Gy.Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021. Section 4.6.2, (4.72), p.223 and Section 4.4, (4.23), p.198.

See Also

Other Hermite Polynomials: HermiteCoeff(), HermiteCov12(), HermiteN()

Examples

```
# Univariate example
H_N_x <- c(1, 2, 3, 4)
x_powers <- HermiteN2X(Type = "Univariate", H_N = H_N_x, N = 4, Sig2 = 1)

# Multivariate example
x <- c(1, 3)
Sig2 <- diag(length(x))
N <- 4
H_N_X <- HermiteN(x, N, Type="Multivariate")
x_ad_n <- HermiteN2X(Type = "Multivariate", H_N = H_N_X, N = N, Sig2 = Sig2)
```

Mom2Cum *Convert moments to cumulants (univariate and multivariate)*

Description

Obtains a vector of cumulants from a vector of moments for either univariate or multivariate data.

Usage

```
Mom2Cum(moments, Type = c("Univariate", "Multivariate"))
```

Arguments

`moments`	Either a vector of univariate moments or a list of vectors of multivariate moments.
`Type`	A character string specifying the type of moments provided. Use "Univariate" for univariate moments and "Multivariate" for multivariate moments.

Value

The vector of cumulants if `Type` is "Univariate" or the list of vectors of cumulants if `Type` is "Multivariate".

References

Gy. Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021. Section 3.4.

See Also

Other Moments and cumulants: Cum2Mom(), EVSKSkewNorm(), EVSKUniS(), MomCumCFUSN(), MomCumSkewNorm(), MomCumUniS(), MomCumZabs()

Examples

```
# Univariate example
mu_x <- c(1, 2, 3, 4)
Mom2Cum(mu_x, Type = "Univariate")

# Multivariate example
mu <- list(c(0,0), c(1,0,0,1), c(0,0,0,0,0,0,0,0), c(3,0,0,1,0,1,1,0,0,1,1,0,1,0,0,3), c(rep(0,32)))
Mom2Cum(mu, Type = "Multivariate")
```

MomCumCFUSN *Moments and cumulants CFUSN*

Description

Provides the theoretical cumulants of the multivariate Canonical Fundamental Skew Normal distribution

Usage

```
MomCumCFUSN(r, d, p, Delta, nMu = FALSE)
```

Arguments

r	The highest cumulant order
d	The multivariate dimension and number of rows of the skewness matrix Delta
p	The number of cols of the skewness matrix Delta
Delta	The skewness matrix
nMu	If set to TRUE, the list of the first r d-variate moments is provided

Value

The list of theoretical cumulants in vector form

References

Gy.Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021, Lemma 5.3 p.251

See Also

Other Moments and cumulants: Cum2Mom(), EVSKSkewNorm(), EVSKUniS(), Mom2Cum(), MomCumSkewNorm(), MomCumUniS(), MomCumZabs()

Examples

```
r <- 4; d <- 2; p <- 3
Lamd <-  matrix(sample(1:50-25, d*p), nrow=d)
ieg<- eigen(diag(p)+t(Lamd)%*%Lamd)
V <- ieg$vectors
Delta <-Lamd %*% V %*% diag(1/sqrt(ieg$values)) %*% t(V)
MomCum <- MomCumCFUSN(r,d,p,Delta)
```

MomCumSkewNorm *Moments and cumulants d-variate Skew Normal*

Description

Computes the theoretical values of moments and cumulants up to the r-th order. Warning: if nMu = TRUE it can be very slow

Usage

```
MomCumSkewNorm(r = 4, omega, alpha, nMu = FALSE)
```

Arguments

r	the highest moment and cumulant order
omega	A $d \times d$ correlation matrix
alpha	shape parameter d-vector
nMu	if it is TRUE then moments are calculated as well

Value

A list of theoretical moments and cumulants

References

Gy.Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021 (5.5) p.247, Lemma 5.1 p. 246

S. R. Jammalamadaka, E. Taufer, Gy. Terdik. On multivariate skewness and kurtosis. Sankhya A, 83(2), 607-644.

See Also

Other Moments and cumulants: Cum2Mom(), EVSKSkewNorm(), EVSKUniS(), Mom2Cum(), MomCumCFUSN(), MomCumUniS(), MomCumZabs()

Examples

```
alpha<-c(10,5,0)
omega<-diag(3)
MomCumSkewNorm(r=4,omega,alpha)
```

`MomCumUniS` *Moments and cumulants Uniform Distribution on the Sphere*

Description

By default, only moments are provided

Usage

```
MomCumUniS(r, d, nCum = FALSE)
```

Arguments

`r`	highest order of moments and cumulants
`d`	dimension
`nCum`	if it is TRUE then cumulants are calculated

Value

The list of moments and cumulants in vector form

References

Gy.Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021 Proposition 5.3 p.297

See Also

Other Moments and cumulants: `Cum2Mom()`, `EVSKSkewNorm()`, `EVSKUniS()`, `Mom2Cum()`, `MomCumCFUSN()`, `MomCumSkewNorm()`, `MomCumZabs()`

Examples

```
# The first four moments for d=3
MomCumUniS(4,3,nCum=0)
# The first four moments and cumulants for d=3
MomCumUniS(4,3,nCum=4)
```

`MomCumZabs` *Moments and Cumulants of the Central Folded Normal Distribution*

Description

Provides the theoretical moments and cumulants of the Central Folded Normal distribution. Depending on the choice of ‘Type‘, either the univariate or d-variate distribution is used.

Usage

```
MomCumZabs(r, d, Type, nCum = FALSE)
```

Arguments

`r`	The highest moment (cumulant) order.
`d`	Integer; the dimension of the distribution. Must be 1 when ‘Type‘ is "Univariate" and greater than 1 when ‘Type‘ is "Multivariate".
`Type`	Character; specifies the type of distribution. Must be either "Univariate" or "Multivariate".
`nCum`	Logical; if TRUE, then cumulants are calculated.

Value

A list containing moments and optionally cumulants.

- For "Univariate" type:
  - `MuZ`: The moments of the univariate Central Folded Normal distribution.
  - `CumZ`: The cumulants of the univariate Central Folded Normal distribution.
- For "Multivariate" type:
  - `MuZ`: The moments of the d-variate Central Folded Normal distribution.
  - `CumZ`: The cumulants of the d-variate Central Folded Normal distribution.

References

Gy.Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021, Proposition 5.1 p.242 and formula: p. 301

See Also

Other Moments and cumulants: `Cum2Mom()`, `EVSKSkewNorm()`, `EVSKUniS()`, `Mom2Cum()`, `MomCumCFUSN()`, `MomCumSkewNorm()`, `MomCumUniS()`

Examples

```
# Univariate case: The first three moments
MomCumZabs(3, 1, Type = "Univariate")
# Univariate case: The first three moments and cumulants
MomCumZabs(3, 1, Type = "Univariate",nCum = TRUE)
# d-variate case: The first three moments
MomCumZabs(3, 2, Type = "Multivariate" )
# d-variate case: The first three moments and cumulants
MomCumZabs(3, d=2, Type = "Multivariate",  nCum = TRUE)
```

`MVStandardize` *Standardize multivariate data*

Description

For data formed by d-variate vectors x with sample covariance S and sample mean M, it computes the values $z = S^{-1/2}(x - M)$

Usage

```
MVStandardize(x)
```

Arguments

`x`	a multivariate data matrix, sample size is the number of rows

Value

a matrix of multivariate data with null mean vector and identity sample covariance matrix

Examples

```
x<-MASS::mvrnorm(1000,c(0,0,1,3),diag(4))
z<-MVStandardize(x)
mu_z<- apply(z,2,mean)
cov_z<- cov(z)
```

Partitions *General Partition Function*

Description

A unified function to compute different types of partitions. Depending on the partition type specified, it calls the appropriate function: Partition_2Perm, Partition_DiagramsClosedNoLoops, Partition_Indecomposable, or Partition_Pairs.

Usage

```
Partitions(Type, ...)
```

Arguments

`Type`	A character string specifying the type of partion to compute. Choose from "2Perm", "Diagram", "Indecomp", "Pairs".
`...`	Additional arguments passed to the specific partition function: **For "2Perm", "Diagram" and "Indecomp":** • `L`: A partition matrix. **For "Pairs":** • `N`: An integer specifying the number of elements to be partitioned.

Value

Depending on the commutator type:

2Perm A vector with the elements 1 to N permuted according to L.

Diagram The list of partition matrices indecomposable with respect to L, representing diagrams without loops.

Indecomp A list of partition matrices indecomposable with respect to L and a vector indicating the number of indecomposable partitions by sizes.

Pairs The list of partition matrices with blocks containing two elements. The list is empty if N is odd.

See Also

Other Partitions: `PartitionTypeAll()`, `PermutationInv()`

Examples

```
# Example for 2Perm
PA <- PartitionTypeAll(4)
Partitions("2Perm", L = PA$Part.class[[3]])

# Example for Diagram
L <- matrix(c(1,1,0,0,0,0,1,1),2,4,byrow=TRUE)
Partitions("Diagram", L = L)
```

```
# Example for Indecomp
L <- matrix(c(1,1,0,0,0,0,1,1),2,4,byrow=TRUE)
Partitions("Indecomp", L = L)

# Example for Pairs
Partitions("Pairs", N = 4)
```

PartitionTypeAll *Partitions, type and number of partitions*

Description

Generates all partitions of `N` numbers and classify them by type

Usage

```
PartitionTypeAll(N)
```

Arguments

`N`	The (integer) number of elements to be partitioned

Value

`Part.class` The list of all possible partitions given as partition matrices

`S_N_r` A vector with the number of partitions of size r=1, r=2, etc. (Stirling numbers of second kind)

`eL_r` A list of partition types with respect to partitions of size r=1, r=2, etc.

`S_r_j` Vectors of number of partitions with given types grouped by partitions of size r=1, r=2, etc.

References

Gy. Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021. Case 1.4, p.31 and Example 1.18, p.32.

See Also

Other Partitions: `Partitions()`, `PermutationInv()`

Examples

```
# See Example 1.18, p. 32, reference below
PTA<-PartitionTypeAll(4)
# Partitions generated
PTA$Part.class
# Partitions of size 2 includes two types
PTA$eL_r[[2]]
# Number of partitions with r=1 blocks, r=2 blocks, etc-
PTA$S_N_r
# Number of different types collected by  partitions of size r=1, r=2, etc.
PTA$S_r_j
# Partitions with size r=2, includes two types (above) each with number
PTA$S_r_j[[2]]
```

PermutationInv *Inverse of a Permutation*

Description

Inverse of a Permutation

Usage

```
PermutationInv(permutation0)
```

Arguments

`permutation0`	A permutation of numbers 1:n

Value

A vector containing the inverse permutation of `permutation0`

References

Gy. Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021, Remark 1.1, p.2

See Also

Other Partitions: `PartitionTypeAll()`, `Partitions()`

`QplicIndx` *Qplication vector*

Description

Restores the duplicated/q-plicated elements which are eliminated by EliminMatr or EliminIndx in a T-product of vectors of dimension d. It produces the same results as QplicMatr.

Usage

```
QplicIndx(d, q)
```

Arguments

d	dimension of the vectors in the T-product
q	power of the Kronecker product

Value

A vector (T-vector) with all elements previously eliminated by EliminIndx

References

Gy. Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021, p.21, (1.31)

See Also

Other Matrices and commutators: `EliminIndx()`, `EliminMatr()`, `QplicMatr()`, `SymIndx()`, `SymMatr()`, `UnivMomCum()`

Examples

```
x<-c(1,2,3)
y<-kronecker(kronecker(x,x),x)
## Distinct elements of y
z<-y[EliminIndx(3,3)]
## Restore eliminated elements in z
z[QplicIndx(3,3)]
```

QplicMatr *Qplication Matrix*

Description

Restores the duplicated/q-plicated elements which are eliminated by EliminMatr in a T-product of vectors of dimension d.

Usage

```
QplicMatr(d, q, useSparse = FALSE)
```

Arguments

`d`	dimension of a vector x
`q`	power of the Kronecker product
`useSparse`	TRUE or FALSE.

Details

Note: since the algorithm of elimination is not unique, q-plication works together with the function EliminMatr only.

Value

Qplication matrix of order $d^q \times \eta_{d,q}$, see (1.30), p.15. If `useSparse=TRUE` an object of the class "dgCMatrix" is produced.

References

Gy. Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021, p.21, (1.31)

See Also

Other Matrices and commutators: `EliminIndx()`, `EliminMatr()`, `QplicIndx()`, `SymIndx()`, `SymMatr()`, `UnivMomCum()`

Examples

```
x<-c(1,2,3)
y<-kronecker(kronecker(x,x),x)
## Distinct elements of y
z<-as.matrix(EliminMatr(3,3))%*%y
## Restore eliminated elements in z
as.vector(QplicMatr(3,3)%*%z)
```

rCFUSN *Random multivariate CFUSN*

Description

Generate random d-vectors from the multivariate Canonical Fundamental Skew-Normal (CFUSN) distribution

Usage

```
rCFUSN(n, Delta)
```

Arguments

n	The number of variates to be generated
Delta	Correlation matrix, the skewness matrix Delta

Value

A random matrix $n \times d$

References

Gy.Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021 (5.5) p.247

S. R. Jammalamadaka, E. Taufer, Gy. Terdik. On multivariate skewness and kurtosis. Sankhya A, 83(2), 607-644.

See Also

Other Random generation: rCFUSSD(), rSkewNorm(), rUniS()

Examples

```
d <- 2; p <- 3
Lamd <-  matrix(sample(1:50-25, d*p), nrow=d)
ieg<- eigen(diag(p)+t(Lamd)%*%Lamd)
V <- ieg$vectors
Delta <-Lamd %*% V %*% diag(1/sqrt(ieg$values)) %*% t(V)
x<-rCFUSN(20,Delta)
```

rCFUSSD *Random multivariate CFUSSD*

Description

Generate random d-vectors from the multivariate Canonical Fundamental Skew-Spherical distribution (CFUSSD) with Gamma generator

Usage

```
rCFUSSD(n, d, p, a, b, Delta)
```

Arguments

`n`	sample size
`d`	dimension
`p`	dimension of the first term of (5.5)
`a`	shape parameter of the Gamma generator
`b`	scale parameter of the Gamma generator
`Delta`	skewness matrix

Value

A matrix of $n \times d$ random numbers

References

Gy.Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021, (5.36) p. 266, (see p.247 for Delta)

See Also

Other Random generation: `rCFUSN()`, `rSkewNorm()`, `rUniS()`

Examples

```
n <- 10^3; d <- 2; p <- 3 ; a <- 1; b <- 1
Lamd <-  matrix(sample(1:50-25, d*p), nrow=d)
ieg<- eigen(diag(p)+t(Lamd)%*%Lamd)
V <- ieg$vectors
Delta <-Lamd %*% V %*% diag(1/sqrt(ieg$values)) %*% t(V)
rCFUSSD(20,d,p,1,1,Delta)
```

rSkewNorm *Random Multivariate Skew Normal*

Description

Generate random d-vectors from the multivariate Skew Normal distribution

Usage

```
rSkewNorm(n, omega, alpha)
```

Arguments

n	sample size
omega	correlation matrix with d dimension
alpha	shape parameter vector of dimension d

Value

A random matrix $n \times d$

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). The Skew-Normal and Related Families. Cambridge University Press, IMS Monographs series.

Gy.H.Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021, Section 5.1.2

See Also

Other Random generation: rCFUSN(), rCFUSSD(), rUniS()

Examples

```
alpha<-c(10,5,0)
omega<-diag(3)
x<-rSkewNorm(20,omega,alpha)
```

rUniS — *Random Uniform on the sphere*

Description

Generate random d-vectors from the Uniform distribution on the sphere

Usage

```
rUniS(n, d)
```

Arguments

n	sample size
d	dimension

Value

A random matrix $n \times d$

References

Gy.Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021

S. R. Jammalamadaka, E. Taufer, Gy. Terdik. On multivariate skewness and kurtosis. Sankhya A, 83(2), 607-644.

See Also

Other Random generation: `rCFUSN()`, `rCFUSSD()`, `rSkewNorm()`

SampleEVSK — *Estimation of multivariate Mean, Variance, T-Skewness and T-Kurtosis vectors*

Description

Provides estimates of mean, variance, skewness and kurtosis vectors for d-variate data

Usage

```
SampleEVSK(X)
```

Arguments

X	d-variate data vector

Value

The list of the estimated mean, variance, skewness and kurtosis vectors

References

Gy.Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021, Sections 6.4.1 and 6.5.1

See Also

Other Estimation: `SampleKurt()`, `SampleMomCum()`, `SampleSkew()`, `VarianceKurt()`, `VarianceSkew()`

Examples

```
x<- MASS::mvrnorm(100,rep(0,3), 3*diag(rep(1,3)))
EVSK<-SampleEVSK(x)
names(EVSK)
EVSK$estSkew
```

`SampleGC` *Gram-Charlier approximation to a multivariate density*

Description

Provides the truncated Gram-Charlier approximation to a multivariate density. Approximation can be up to the first k=8 cumulants.

Usage

```
SampleGC(X, k = 4, cum = NULL)
```

Arguments

`X`	A matrix of d-variate data
`k`	the order of the approximation, by default set to 4; (k must not be smaller than 3 or greater than 8)
`cum`	if NULL (default) the cumulant vector is estimated from X. If `cum` is provided no estimation of cumulants is performed.

Value

The vector of the Gram-Charlier density evaluated at X

References

Gy.Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021. Section 4.7.

Examples

```
# Gram-Charlier density approximation (k=4) of data generated from
# a bivariate skew-gaussian distribution
n<-50
alpha<-c(10,0)
omega<-diag(2)
X<-rSkewNorm(n,omega,alpha)
EC<-SampleEVSK(X)
fy4<-SampleGC(X[1:5,],cum=EC)
```

SampleHermiteN *Estimate the N-th d-variate Hermite polynomial*

Description

The vector x is standardized and the N-th d-variate polynomial is computed

Usage

```
SampleHermiteN(x, N)
```

Arguments

x	a d-variate data vector
N	the order of the d-variate Hermite polynomial

Value

The vector of the N-th d-variate polynomial

Examples

```
x<-MASS::mvrnorm(100,rep(0,3),diag(3))
H3<-SampleHermiteN(x,3)
```

SampleKurt *Estimation of Sample Kurtosis (Mardia, MRSz, Total)*

Description

Estimates the sample kurtosis index based on the specified method: Mardia, MRSz, or Total.

Usage

```
SampleKurt(x, Type = c("Mardia", "MRSz", "Total"))
```

Arguments

x	A matrix of multivariate data.
Type	A character string specifying the type of kurtosis index to estimate. Use "Mardia" for Mardia's kurtosis index, "MRSz" for the Mori-Rohatgi-Szekely kurtosis matrix, or "Total" for the total kurtosis index.

Value

A list containing the estimated kurtosis index or matrix and the associated p-value under the Gaussian hypothesis.

Mardia.Kurtosis	The kurtosis index when Type is "Mardia".
MRSz.Kurtosis	The kurtosis matrix when Type is "MRSz".
Total.Kurtosis	The total kurtosis index when Type is "Total".
p.value	The p-value under the Gaussian hypothesis for the estimated kurtosis.

References

Gy.Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021. Example 6.1 and 6.9.

See Also

Other Estimation: SampleEVSK(), SampleMomCum(), SampleSkew(), VarianceKurt(), VarianceSkew()

Examples

```
# Mardia's kurtosis example
x <- matrix(rnorm(100*5), ncol=5)
SampleKurt(x, Type = "Mardia")

# MRSz's kurtosis example
SampleKurt(x, Type = "MRSz")

# Total kurtosis example
SampleKurt(x, Type = "Total")
```

SampleMomCum *Estimation of multivariate T-Moments and T-Cumulants*

Description

Provides estimates of univariate and multivariate moments and cumulants up to order r. By default data are standardized; using only demeaned or raw data is also possible.

Usage

```
SampleMomCum(X, r, centering = FALSE, scaling = TRUE)
```

Arguments

X	d-vector data
r	The highest moment order (r >2)
centering	set to T (and scaling = F) if only centering is needed
scaling	set to T (and centering=F) if standardization of multivariate data is needed

Value

estMu.r: the list of the multivariate moments up to order r

estCum.r: the list of the multivariate cumulants up to order r

References

Gy.Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021.

See Also

Other Estimation: SampleEVSK(), SampleKurt(), SampleSkew(), VarianceKurt(), VarianceSkew()

Examples

```
## generate random data from a 3-variate skew normal distribution
alpha<-c(10,5,0)
omega<-diag(3)
x<-rSkewNorm(50,omega,alpha)
## estimate the first three moments and cumulants from raw (uncentered and unstandardized) data
SampleMomCum(x,3,centering=FALSE,scaling=FALSE)
## estimate the first three moments and cumulants from standardized data
SampleMomCum(x,3,centering=FALSE,scaling=TRUE)
```

SampleSkew — *Estimation of Sample Skewness (Mardia, MRSz)*

Description

Estimates the sample skewness index based on the specified method: Mardia or MRSz.

Usage

```
SampleSkew(x, Type = c("Mardia", "MRSz"))
```

Arguments

`x` A matrix of multivariate data.

`Type` A character string specifying the type of skewness index to estimate. Use "Mardia" for Mardia's skewness index or "MRSz" for the Mori-Rohatgi-Szekely skewness vector and index.

Value

A list containing the estimated skewness index or vector and the associated p-value under the Gaussian hypothesis.

`Mardia.Skewness` The skewness index when `Type` is "Mardia".

`MRSz.Skewness.Vector` The skewness vector when `Type` is "MRSz".

`MRSz.Skewness.Index` The skewness index when `Type` is "MRSz".

`p.value` The p-value under the Gaussian hypothesis for the estimated skewness.

References

Gy.Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021. Example 6.1 and 6.2.

S. R. Jammalamadaka, E. Taufer, Gy. Terdik. On multivariate skewness and kurtosis. Sankhya A, 83(2), 607-644.

See Also

Other Estimation: `SampleEVSK()`, `SampleKurt()`, `SampleMomCum()`, `VarianceKurt()`, `VarianceSkew()`

Examples

```
# Mardia's skewness example
x <- matrix(rnorm(100*5), ncol=5)
SampleSkew(x, Type = "Mardia")

# MRSz's skewness example
SampleSkew(x, Type = "MRSz")
```

`SampleVarianceSkewKurt` *Estimated Variance of skewness and kurtosis vectors*

Description

Provides the estimated covariance matrices of the data-estimated skewness and kurtosis vectors.

Usage

```
SampleVarianceSkewKurt(X)
```

Arguments

X	A matrix of d-variate data

Value

The list of covariance matrices of the skewness and kurtosis vectors

References

Gy.Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021.

`SymIndx` *Symmetrizing vector*

Description

Vector symmetrizing a T-product of vectors of the same dimension d. Produces the same results as SymMatr

Usage

```
SymIndx(x, d, n)
```

Arguments

x	the vector to be symmetrized of dimension d^n
d	size of the single vectors in the product
n	power of the T-product

Value

A vector with the symmetrized version of x of dimension d^n

References

Gy. Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021.Section 1.3.1 Symmetrization, p.14. (1.29)

See Also

Other Matrices and commutators: EliminIndx(), EliminMatr(), QplicIndx(), QplicMatr(), SymMatr(), UnivMomCum()

Examples

```
a<-c(1,2)
b<-c(2,3)
c<-kronecker(kronecker(a,a),b)
## The symmetrized version of c is
SymIndx(c,2,3)
```

SymMatr — *Symmetrizer Matrix*

Description

Based on Chacon and Duong (2015) efficient recursive algorithms for functionals based on higher order derivatives. An option for sparse matrix is provided. By using sparse matrices far less memory is required and faster computation times are obtained

Usage

```
SymMatr(d, n, useSparse = FALSE)
```

Arguments

d	dimension of a vector x
n	power of the Kronecker product
useSparse	TRUE or FALSE. If TRUE an object of the class "dgCMatrix" is produced.

Value

A Symmetrizer matrix with order $d^n \times d^n$. If `useSparse=TRUE` an object of the class "dgCMatrix" is produced.

References

Chacon, J. E., and Duong, T. (2015). Efficient recursive algorithms for functionals based on higher order derivatives of the multivariate Gaussian density. Statistics and Computing, 25(5), 959-974.

Gy. Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021.Section 1.3.1 Symmetrization, p.14. (1.29)

See Also

Other Matrices and commutators: EliminIndx(), EliminMatr(), QplicIndx(), QplicMatr(), SymIndx(), UnivMomCum()

Examples

```
a<-c(1,2)
b<-c(2,3)
c<-kronecker(kronecker(a,a),b)
## The symmetrized version of c is
as.vector(SymMatr(2,3)%*%c)
```

UnivMomCum *Univariate moments and cumulants from T-vectors*

Description

A vector of indexes to select the moments and cumulants of the single components of the random vector X for which a T-vector of moments and cumulants is available

Usage

```
UnivMomCum(d, q)
```

Arguments

d	dimension of a vector X
q	power of the Kronecker product

Value

A vector of indexes

See Also

Other Matrices and commutators: EliminIndx(), EliminMatr(), QplicIndx(), QplicMatr(), SymIndx(), SymMatr()

Examples

```
## For a 3-variate skewness and kurtosis vectors estimated from data, extract
## the skewness and kurtosis estimates for each of the single components of the vector
alpha<-c(10,5,0)
omega<-diag(rep(1,3))
X<-rSkewNorm(200, omega, alpha)
EVSK<-SampleEVSK(X)
## Get the univariate skewness and kurtosis for X1,X2,X3
EVSK$estSkew[UnivMomCum(3,3)]
EVSK$estKurt[UnivMomCum(3,4)]
```

`VarianceKurt` *Asymptotic Variance of the estimated kurtosis vector*

Description

Warning: the function requires 8! computations, for d>3, the timing required maybe large.

Usage

```
VarianceKurt(cum)
```

Arguments

`cum`	The theoretical/estimated cumulants up to the 8th order in vector form

Value

The matrix of theoretical/estimated variance

References

Gy.Terdik, Multivariate statistical methods - going beyond the linear, Springer 2021. Ch. 6, formula (6.26)

See Also

Other Estimation: `SampleEVSK()`, `SampleKurt()`, `SampleMomCum()`, `SampleSkew()`, `VarianceSkew()`

`VarianceSkew` *Asymptotic Variance of the estimated skewness vector*

Description

Asymptotic Variance of the estimated skewness vector

Usage

```
VarianceSkew(cum)
```

Arguments

`cum`	The theoretical/estimated cumulants up to order 6 in vector form

Value

The matrix of theoretical/estimated variance

References

Gy.Terdik, Multivariate statistical methods - Going beyond the linear, Springer 2021. Ch.6, formula (6.13)

See Also

Other Estimation: SampleEVSK(), SampleKurt(), SampleMomCum(), SampleSkew(), VarianceKurt()

Examples

```
alpha<-c(10,5)
omega<-diag(rep(1,2))
MC <- MomCumSkewNorm(r = 6,omega,alpha)
cum <- MC$CumX
VS <- VarianceSkew(cum)
```

Index