

# Package: covKCD (via r-universe)

September 5, 2024

**Title** Covariance Estimation for Matrix Data with the Kronecker-Core Decomposition

**Version** 0.1

**Description** Matrix-variate covariance estimation via the Kronecker-core decomposition. Computes the Kronecker and core covariance matrices corresponding to an arbitrary covariance matrix, and provides an empirical Bayes covariance estimator that adaptively shrinks towards the space of separable covariance matrices. For details, see Hoff, McCormack and Zhang (2022) <[arXiv:2207.12484](https://arxiv.org/abs/2207.12484)> ``Core Shrinkage Covariance Estimation for Matrix-variate data".

**Date** 2022-08-10

**License** GPL-3

**RoxygenNote** 7.2.1

**Repository** <https://pdhoff.r-universe.dev>

**RemoteUrl** <https://github.com/pdhoff/covkcd>

**RemoteRef** HEAD

**RemoteSha** 19ae9da1166d55f4fa9ad111364b66aec3a70b81

## Contents

ca2cm . . . . .	2
cm2ca . . . . .	2
covCSE . . . . .	3
covKCD . . . . .	4
lmvgamma . . . . .	5
mcov . . . . .	6
msqrt . . . . .	7
msqrtInv . . . . .	7

**Index** [9](#)

---

ca2cm	<i>Covariance array to covariance matrix</i>
-------	--

---

**Description**

Reshape a covariance array to a covariance matrix.

**Usage**

```
ca2cm(A)
```

**Arguments**

A                    a covariance array of dimension  $p1 * p2 * p1 * p2$ .

**Value**

a  $p1 * p2$  by  $p1 * p2$  covariance matrix.

**Author(s)**

Peter Hoff

**Examples**

```
p1<-4 ; p2<-7 ; p<-p1*p2
S<-rWishart(1,p,diag(p))[, ,1]
A<-cm2ca(S,p1,p2)
range(S-ca2cm(A))
```

---

cm2ca	<i>Covariance matrix to covariance array</i>
-------	--

---

**Description**

Reshape a covariance matrix to a covariance array.

**Usage**

```
cm2ca(S, p1, p2)
```

**Arguments**

S                    a covariance matrix of dimension  $(p1 * p2) * (p1 * p2)$ .  
 p1                    the row dimension.  
 p2                    the column dimension.

**Value**

a four-way array where entry  $i1,j1,i2,j2$  gives the covariance between element  $i1,j1$  and element  $i2,j2$  of a random matrix.

**Author(s)**

Peter Hoff

**Examples**

```
p1<-4 ; p2<-7 ; p<-p1*p2
S<-rWishart(1,p,diag(p))[, , 1]
A<-cm2ca(S,p1,p2)
range(S-ca2cm(A))
```

---

 covCSE

*Empirical Bayes core shrinkage covariance estimator*


---

**Description**

Estimate a covariance matrix by adaptively shrinking the core.

**Usage**

```
covCSE(data, n = NULL, p1 = NULL, p2 = NULL, tol = 1e-08)
```

**Arguments**

data	either a numeric $n \times p1 \times p2$ array consisting of $n$ data matrices each of dimension $p1 \times p2$ , or a $p1 \times p2$ covariance matrix of data of this type. If the latter, the values of $n$ , $p1$ and $p2$ must be specified.
n	the sample size.
p1	the row dimension of the data matrices.
p2	the column dimension of the data matrices.
tol	the convergence tolerance of the iterative algorithm.

**Value**

a covariance matrix of the same dimension as  $S$ . The attribute  $w$  of  $S$  gives the shrinkage weight on the Kronecker covariance of  $S$ .

**Author(s)**

Peter Hoff

**Examples**

```

p1<-4 ; p2<-3 ; n<-20

# create a matrix Y with separable covariance
Sig1<-rWishart(1,p1,diag(p1))[,,1]
Sig2<-rWishart(1,p2,diag(p2))[,,1]

Y<-array(rnorm(n*p1*p2),dim=c(n,p1,p2))
Y<-aperm( apply(Y,c(1,3),function(y){ msqrt(Sig1)%*%y } ),c(2,1,3))
Y<-aperm( apply(Y,c(1,2),function(y){ msqrt(Sig2)%*%y } ),c(2,3,1))

# covariance
S<-mcov(Y)
covCSE(S,n,p1,p2)

# now an unstructured covariance
S<-rWishart(1,p1*p2,diag(p1*p2))[,,1]
covCSE(S,n,p1,p2)

```

covKCD

*Kronecker-core covariance decomposition***Description**

Computes the Kronecker-core decomposition of a covariance matrix.

**Usage**

```
covKCD(S, p1, p2, tol = 1e-08)
```

**Arguments**

S	a covariance matrix of dimension (p1p2)*(p1p2).
p1	the row dimension.
p2	the column dimension.
tol	the convergence tolerance of the iterative algorithm.

**Details**

The Kronecker-core decomposition is a representation of an arbitrary covariance matrix  $S$  in terms of a separable Kronecker covariance matrix  $K$  and a complementary non-separable core covariance matrix  $C$ . The Kronecker covariance is the separable covariance matrix that is closest to  $S$  in terms of the divergence function

$$\log |K| + \text{trace}(K^{-1}S).$$

The core covariance matrix  $C$  is computed from  $S$  and  $K$  via

$$C = K^{-1/2}SK^{-1/2}.$$

**Value**

covKCD returns a list with the following elements:

**K** the Kronecker covariance matrix;

**C** the core covariance matrix;

**K1** the row covariance matrix;

**K2** the column covariance matrix;

**div** the divergence between S and K across iterations of the algorithm.

**Author(s)**

Peter Hoff

**Examples**

```
p1<-4 ; p2<-3 ; n<-200

# create a matrix Y with separable covariance
A<-matrix(rnorm(p1*p1),p1,p1)
B<-matrix(rnorm(p2*p2),p2,p2)/3
Y<-array(rnorm(n*p1*p2),dim=c(n,p1,p2))
Y<-aperm( apply(Y,c(1,3),function(y){ A**y } ),c(2,1,3))
Y<-aperm( apply(Y,c(1,2),function(y){ B**y } ),c(2,3,1))

# covariance
S<-mcov(Y)

KCD<-covKCD(S,p1,p2)

plot(A**t(A), KCD$K1)
plot(B**t(B), KCD$K2)
```

---

lmgamma

*Log multivariate gamma function*


---

**Description**

Compute the logarithm of the multivariate gamma function  $\log \Gamma_p(a)$ .

**Usage**

```
lmgamma(a, p)
```

**Arguments**

**a** a numeric scalar.  
**p** a positive integer.

**Value**

a scalar

**Author(s)**

Peter Hoff

---

mcov

*Matrix-variate covariance matrix*

---

**Description**

Compute the covariance matrix of a sample of data matrices.

**Usage**

```
mcov(Y, use = "everything")
```

**Arguments**

**Y** a numeric  $n \times p1 \times p2$  data array corresponding to  $n$  data matrices of dimension  $p1 \times p2$ .

**use** a character string giving method for dealing with missing values, fed to the `cov` function.

**Value**

a  $p1 \times p2$  by  $p1 \times p2$  sample covariance matrix of the  $n$  vectorized data matrices.

**Author(s)**

Peter Hoff

**Examples**

```
p1<-4 ; p2<-3 ; n<-200

# create a matrix Y with separable covariance
Sig1<-rWishart(1,p1,diag(p1))[, , 1]
Sig2<-rWishart(1,p2,diag(p2))[, , 1]

Y<-array(rnorm(n*p1*p2),dim=c(n,p1,p2))
Y<-aperm( apply(Y,c(1,3),function(y){ msqrt(Sig1)%*%y } ),c(2,1,3))
Y<-aperm( apply(Y,c(1,2),function(y){ msqrt(Sig2)%*%y } ),c(2,3,1))

# covariance
S<-mcov(Y)
image(S)
```

```
plot(S,kronecker(Sig2,Sig1)) ; abline(0,1)
```

---

msqrt

*Symmetric square root of a matrix*

---

**Description**

Compute the symmetric square root of a matrix.

**Usage**

```
msqrt(M)
```

**Arguments**

M                    a positive semidefinite matrix.

**Value**

a positive semidefinite matrix.

**Author(s)**

Peter Hoff

**Examples**

```
S<-rWishart(1,5,diag(5))[, ,1]
S
Sh<-msqrt(S)
Sh**%Sh
```

---

msqrtInv

*Inverse symmetric square root of a matrix*

---

**Description**

Compute the inverse of the symmetric square root of a matrix.

**Usage**

```
msqrtInv(M)
```

**Arguments**

M a positive definite matrix.

**Value**

a positive definite matrix.

**Author(s)**

Peter Hoff

**Examples**

```
S<-rWishart(1,5,diag(5))[, ,1]
solve(S)
iSh<-msqrtInv(S)
iSh%*%iSh
```



# Index

ca2cm, [2](#)

cm2ca, [2](#)

cov, [6](#)

covCSE, [3](#)

covKCD, [4](#)

lmgamma, [5](#)

mcov, [6](#)

msqrt, [7](#)

msqrtInv, [7](#)