

# Package ‘vstdct’

May 8, 2026

**Title** Nonparametric Estimation of Toeplitz Covariance Matrices

**Version** 0.2

**Description** A nonparametric method to estimate Toeplitz covariance matrices from a sample of  $n$  independently and identically distributed  $p$ -dimensional vectors with mean zero. The data is pre-processed with the discrete cosine matrix and a variance stabilization transformation to obtain an approximate Gaussian regression setting for the log-spectral density function. Estimates of the spectral density function and the inverse of the covariance matrix are provided as well. Functions for simulating data and a protein data example are included. For details see (Klockmann, Krivobokova; 2023), <[doi:10.48550/arXiv.2303.10018](https://doi.org/10.48550/arXiv.2303.10018)>.

**License** GPL-2

**Encoding** UTF-8

**RoxygenNote** 7.2.1

**Imports** dtt, MASS, nlme

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Depends** R (>= 3.5.0)

**LazyData** true

**LazyDataCompression** xz

**NeedsCompilation** no

**Author** Karolina Klockmann [aut, cre],  
Tatyana Krivobokova [aut]

**Maintainer** Karolina Klockmann <[karolina.klockmann@gmx.de](mailto:karolina.klockmann@gmx.de)>

**Repository** CRAN

**Date/Publication** 2023-07-06 07:30:02 UTC

## Contents

aquaporin . . . . .	2
Data Examples . . . . .	2
Data.trafo . . . . .	3
DR.basis . . . . .	4
Toep.estimator . . . . .	5

**Index**[7](#)

aquaporin

*Aquaporin Dataset***Description**

Dataset with molecular dynamics simulations for the yeast aquaporin (Aqy1) - the gated water channel of the yeast *Pichi pastoris*. The dataset contains only the diameter  $Y$  of the channel which is used in the data analysis in (Klockmann and Krivobokova, 2023). The diameter  $Y$  is measured by the distance between two centers of mass of certain residues of the protein. The dataset includes a 100 nanosecond time frame, split into 20000 equidistant observations. The full dataset, including the Euclidean coordinates of all 783 atoms, is available from the authors. For more details see (Klockmann, Krivobokova; 2023).

**Usage**

```
aquaporin
```

**Format**

A data frame with 20000 rows and 1 variable:

- $Y$ : the diameter of the channel

**Source**

see (Klockmann, Krivobokova; 2023).

**Examples**

```
data(aquaporin)
```

Data Examples

*Data Examples***Description**

[example1](#), [example2](#) and [example3](#) generate i.i.d. vectors from a given distribution with different Toeplitz covariance matrices. The covariance function  $\sigma$  of the Toeplitz covariance matrix of

- [example1](#): has a polynomial decay,  $\sigma(\tau) = sd^2(1 + |\tau|)^{-gamma}$ ,
- [example2](#): follows an  $ARMA(2, 2)$  model with coefficients  $(0.7, -0.4, -0.2, 0.2)$  and innovations variance  $sd^2$ ,
- [example3](#): yields a Lipschitz continuous spectral density  $f$  that is not differentiable, i.e.  $f(x) = sd^2(|\sin(x + 0.5\pi)|^{gamma} + 0.45)$

**Usage**

```
example1(p, n, sd, gamma, family = "Gaussian")
```

```
example2(p, n, sd, family = "Gaussian")
```

```
example3(p, n, sd, gamma, family = "Gaussian")
```

**Arguments**

p	vector length
n	sample size
sd	standard deviation
gamma	polynomial decay of covariance function for example1 resp. exponent for example3
family	distribution of the simulated data. Available distributions are "Gaussian", "Gamma", "Uniform". The default is "Gaussian".

**Value**

A list containing the following elements:

- Y: pxn dimensional data matrix
- sdf: true spectral density function
- acf: true covariance function

**Examples**

```
example1(p=10, n=1, sd=1, gamma=1.2, family="Gaussian")
```

```
example2(p=10, n=1, sd=1, family="Gaussian")
```

```
example3(p=10, n=1, sd=1, gamma=2, family="Gaussian")
```

---

 Data.trafo

*Data Transformation*


---

**Description**

Applies the Discrete Cosine I transform, data binning and the variance stabilizing transform function to the data.

**Usage**

```
Data.trafo(y, Te, dct.out = FALSE)
```

**Arguments**

y	n x p dimensional data matrix
Te	number of bins for data binning. Te should be smaller than the vector length p.
dct.out	logical. If TRUE, the p-dim. DCT-I matrix is returned. The default is FALSE.

**Value**

A list containing the following elements:

- `m`: number of data points per bin, that is  $m=n*\text{round}(p/Te)$ . If  $p/Te$  is not an integer, the first/last bin may contain more than  $m$  data points.
- `y.star`:  $2Te-2$  dimensional vector with binned, variance stabilized and mirrored data. The bin number  $Te$  may be modified to guarantee at least two data points per bin. If  $p/Te$  is not an integer, the vector dimension is  $2*\text{floor}(p/\text{round}(p/Te))-2$ .
- `dct.matrix`:  $p$ -dim. DCT-I matrix (if `dct.out=TRUE`)

---

DR.basis

*Periodic Demmler-Reinsch Basis*


---

**Description**

Calculates the periodic Demmler-Reinsch basis for a given smoothness and a given vector of grid points. For details see (Schwarz, Krivobokova; 2016).

**Usage**

```
DR.basis(x, n, q)
```

**Arguments**

<code>x</code>	$m$ -dim. vector with grid values in $[0,1]$
<code>n</code>	dimension of the basis
<code>q</code>	penalization order, $q=1, 2, 3, 4$ are available

**Value**

$m \times n$  dimensional matrix with the  $n$  DR basis functions evaluated at grid points  $x$

**Examples**

```
DR.basis(seq(1,10)/10,5,2)
```

---

Toep.estimator                      *Toeplitz Covariance and Precision Matrix Estimator*

---

### Description

Estimates the Toeplitz covariance matrix, the inverse matrix and the spectral density from a sample of  $n$  i.i.d.  $p$ -dimensional vectors with mean zero.

### Usage

```
Toep.estimator(y, Te, q, method, f.true = NULL)
```

### Arguments

<code>y</code>	$n \times p$ dimensional data matrix
<code>Te</code>	number of bins for data binning.
<code>q</code>	penalization order, $q=1, 2, 3, 4$ are available
<code>method</code>	to select the smoothing parameter of the smoothing spline. Available methods are restricted maximum likelihood "ML", generalized cross-validation "GCV" and the oracle versions "ML-oracle", "GCV-oracle".
<code>f.true</code>	$Te$ -dimensional vector with the true spectral density function evaluated at equispaced points in $[0, \pi]$ . Only required, if an oracle method ("ML-oracle", "GCV-oracle") is chosen for method.

### Value

A list containing the following elements:

- `toep`:  $p$ -dim. Toeplitz covariance matrix
- `toep.inv`:  $p$ -dim. precision matrix
- `acf`:  $p$ -dim. vector with the covariance function
- `sdf`:  $p$ -dim. vector with the spectral density in the interval  $[0, 1]$

### Examples

```
#EXAMPLE 1: Simulate Gaussian ARMA(2,2)
library(nlme)
library(MASS)
p=100
n=1
Sigma=1.44*corMatrix(Initialize(corARMA(c(0.7, -0.4, -0.2, 0.2), p=2, q=2), data=diag(1:p)))
Y=matrix(mvrnorm(n, mu=numeric(p), Sigma=Sigma), n, p)
fit.toep=Toep.estimator(y=Y, Te=10, q=2, method="GCV")$toep
```

```
#EXAMPLE 2: AQUAPORIN DATA
```

```
data(aquaporin)
n=length(aquaporin$Y)
y.train=aquaporin$Y[1:(0.01*n)]
y.train=y.train-mean(y.train)
fit.toep=Toep.estimator(y=y.train,Te=10,q=1,method="ML")$toep
```

# Index

## \* datasets

aquaporin, [2](#)

aquaporin, [2](#)

Data Examples, [2](#)

Data.trafo, [3](#)

DR.basis, [4](#)

example1, [2](#)

example1 (Data Examples), [2](#)

example2, [2](#)

example2 (Data Examples), [2](#)

example3, [2](#)

example3 (Data Examples), [2](#)

Toep.estimator, [5](#)