

# Package: ExtrPatt (via r-universe)

November 2, 2024

**Type** Package

**Title** Spatial Dependencies and Indices for Extremes

**Version** 0.1-4

**Author** Svenja Szemkus [aut, cre], Dan Cooley [ctb], Yuing Jiang [ctb]

**Description** An implementation of 1) the tail pairwise dependence matrix (TPDM) as described in Jiang & Cooley (2020) [doi:10.1175/JCLI-D-19-0413.1](https://doi.org/10.1175/JCLI-D-19-0413.1) > 2) the extremal pattern index (EPI) as described in Szemkus & Friederichs ('Spatial patterns and indices for heatwave and droughts over Europe using a decomposition of extremal dependency'; submitted to ASCMO 2023).

**Depends** R (>= 3.5.0)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** Matrix, doParallel, stats, foreach, MASS, parallel, utils

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Maintainer** Svenja Szemkus <sszemkus@uni-bonn.de>

**Date/Publication** 2023-11-07 18:50:05 UTC

**Repository** <https://svenja12.r-universe.dev>

**RemoteUrl** <https://github.com/cran/ExtrPatt>

**RemoteRef** HEAD

**RemoteSha** 116045d1ecf44fdcd257c61b5fc35a2b7ee6c41

## Contents

compute.EPI . . . . .	2
decls . . . . .	3
est.tpdm . . . . .	4

invTrans . . . . .	5
pca.tpdm . . . . .	6
precipGER . . . . .	7
svd.tpdm . . . . .	7
to.alpha.2 . . . . .	8
trans . . . . .	8
wrapper.EPI . . . . .	9

## Index 11

---

compute.EPI	<i>Estimation of EPI</i>
-------------	--------------------------

---

### Description

Estimates the extremal pattern index (EPI) from either the 'm' principle components after a PCA or left- and right expansion coefficients after an SVD. In case of a SVD, the threshold-based EPI (TEPI) can optionally be calculated.

### Usage

```
compute.EPI(coeff, m = 1:10, q = 0.98)
```

### Arguments

coeff	A list, containing the t x n dimensional principle components/expansion coefficients of TPDM. Can also be output of function 'est.tpdm'.
m	numeric vector: Containing the Principle Components from which EPI shall be computed (e.g. with modes = c(1:10), the EPI is calculated on first ten principle components)
q	Optional: A threshold for computation of TEPI

### Details

Given the first 'm' modes of principle components u and eigenvalues after a PCA, the EPI is given as:

$$EPI_t^u = \sqrt{\sum_{k=1}^m (u_{t,k}^2) / \sum_{j=1}^m e_j}$$

Given the first 'm' modes of expansion coefficients u and v and singular values e after a SVD, the EPI and TEPI are given as:

$$EPI_t^{u,v} = \sqrt{\sum_{k=1}^m (u_{t,k}^2 + v_{t,k}^2) / \sum_{j=1}^m e_j}$$

$$TEPI_t^{u,v} = \sqrt{\left( \sum_{k=1}^m (u_{t,k}^2 + v_{t,k}^2) / \sum_{j=1}^m e_j \right) |_{(|u_{t,k}| > q_u, |v_{t,k}| > q_v)}}$$

**Value**

An array of length  $t$ , containing EPI. TEPI is computed if  $q > 0$ .

**References**

Szemkus & Friederichs (2023)

**Examples**

```
data <- precipGER

data.alpha2 <- to.alpha.2(data$pr)
Sigma <- est.tpdm(data.alpha2,anz_cores =1)
res.pca <- pca.tpdm(Sigma, data.alpha2)
EPI <- compute.EPI(res.pca, m = 1:10)

plot(data$date, EPI, type='l')
```

---

decls

*Declustering*

---

**Description**

Declustering routine, which will can be applied on radial component  $r$  in estimation of the TPDM. Subroutine of [est.tpdm](#).

**Usage**

```
decls(x, th, k)
```

**Arguments**

x	Real vector
th	Threshold
k	Cluster length

**Value**

numeric vector of declustered threshold exceedances

**Author(s)**

Yuing Jiang, Dan Cooley

**References**

Jiang & Cooley (2020) <doi:10.1175/JCLI-D-19-0413.1>

**See Also**[est.tpdm](#)

est.tpdm

*Estimation of TPDM***Description**

Estimation of tail pairwise dependence matrix (TPDM)

Sub-Routine of [est.row.tpdm](#). Calculates one element of the TPDM

**Usage**

```
est.tpdm(X, Y = NULL, anz_cores = 1, clust = NULL, q = 0.98)
```

```
est.row.tpdm(x, Y, clust = NULL, q = 0.98)
```

```
est.element.tpdm(x, y, clust = NULL, q = 0.98)
```

**Arguments**

X	A t x n dimensional, numeric data-matrix with t: Number of time steps and n: Number of grid points/stations
Y	A t x n dimensional, numeric Data-matrix with t: Number of time steps and n: Number of grid points/stations
anz_cores	Number of cores for parallel computing (default:1); Be careful not to overload your computer!
clust	Optional: If clust = NULL, no declustering is performed. Else, declustering according to cluster-length 'clust'.
q	Threshold for computation of TPDM. Only data above the 'q'-quantile will be used for estimation. Choose such that 0<q<1.
x	Array of length t, where t is the number of time steps
y	Same as x

**Details**

Given a random vector X with components  $x_{t,i}, x_{t,j}$  with  $i, j = 1, \dots, n$  and it's radial component  $r_{t,ij} = \sqrt{x_{t,i}^2 + x_{t,j}^2}$  and angular components  $w_{t,i} = x_{t,i}/r_{t,ij}$  and  $w_{t,j} = x_{t,j}/r_{t,ij}$ , the i'th,j'th element of the TPDM is estimated as:

$$\hat{\sigma}_{ij} = 2n_{ij,exc}^{-1} \sum_{t=1}^n w_{t,i} w_{t,j} |_{(r_{t,ij} > r_{0,ij})}$$

. Given two random vectors  $X$  and  $Y$  with components  $x_{t,i}, y_{t,j}$  with  $i, j = 1, \dots, n$ , and it's radial component  $r_{t,ij} = \sqrt{x_{t,i}^2 + y_{t,j}^2}$  and angular components  $w_{t,i}^x = \frac{x_{t,i}}{r_{t,ij}}, w_{t,j}^y = \frac{y_{t,j}}{r_{t,ij}}$ , the  $i$ 'th, $j$ 'th element of the cross-TPDM is estimated as:

$$\hat{\sigma}_{ij} = 2n_{exc}^{-1} \sum_{t=1}^n w_{t,i}^x w_{t,j}^y |_{(r_{t,ij} > r_{0,ij})}$$

### Value

An  $n \times n$  matrix, containing the estimate of the TPDM  
 Array containing the estimate of one row of the TPDM.  
 Value containing the estimate of one element of the TPDM.

### References

Jiang & Cooley (2020) <doi:10.1175/JCLI-D-19-0413.1>; Szemkus & Friederichs (2023)

### Examples

```
data <- precipGER

data.alpha2 <- to.alpha.2(data$pr)
Sigma <- est.tpdm(data.alpha2,anz_cores =1)
```

---

<code>invTrans</code>	<i>Transformation function</i>
-----------------------	--------------------------------

---

### Description

Applies the inverse transformation  $t^{-1}(v) = \log(\exp(v) - 1)$

### Usage

```
invTrans(v)
```

### Arguments

`v` Real, positive vector

### Details

Transformation from real, positive vector in real vector under preservation of frechet-distribution.

### Value

Real vector, containing the result of inverse transformation function.

**Author(s)**

Yuing Jiang, Dan Cooley

**References**

Cooley & Thibaud (2019) <doi:10.1093/biomet/asz028>

**See Also**

[svd.tpdm](#), [pca.tpdm](#)

---

pca. tpdm

*Principal Component Analysis for TPDM*

---

**Description**

Calculates principal component analysis (PCA) of given TPDM

**Usage**

```
pca.tpdm(Sigma, data)
```

**Arguments**

Sigma	A n x n data array, containing the TPDM, can be output of <a href="#">est.tpdm</a> .
data	A t x n dimensional, numeric Data-matrix with t: Number of time steps and n: Number of grid points/stations.

**Value**

list containing

- pc: The Principal Components of TPDM
- basis: The Eigenvectors of TPDM
- extremal.basis: The Eigenvectors of TPDM but transformed in positive reals with [trans](#)

**Author(s)**

Yuing Jiang, Dan Cooley

**References**

Jiang & Cooley (2020) <doi:10.1175/JCLI-D-19-0413.1>

---

```
precipGER
```

```
daily Precipitation over Southern Germany
```

---

**Description**

Daily Precipitation at several stations in Germany

**Usage**

```
data(precipGER)
```

**Format**

A list containing containing

- pr: data-array
- date: time-information
- lon,lat: longitude & latitude information

**Details**

Daily Precipitation Data

Daily precipitation data from several wather station in southern Germany (longitude <50) over the years 2000-2019. The data has been downloaded from opendata server of german weather service ([https://opendata.dwd.de/climate\\_environment/CDC/observations\\_germany/climate/daily/kl/historical/](https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/daily/kl/historical/)).

**Source**

Quelle: Deutscher Wetterdienst

---

```
svd.tpdm
```

```
Singular Value decomposition for cross-TPDM
```

---

**Description**

Calculates singular value decomposition (SVD) of given cross-TPDM

**Usage**

```
svd.tpdm(Sigma, X, Y)
```

**Arguments**

Sigma	A n x n data array, containing the cross-TPDM, can be output of <a href="#">est.tpdm</a> .
X	A t x n dimensional, numeric Data-matrix with t: Number of time steps and n: Number of grid points/stations.
Y	Same as X but for second variable.

**Value**

List containing

- pcU, pcV: The left- and right expansion coefficients of cross-TPDM
- U, V: The left- and right singular Vectors of cross-TPDM
- extr.U, extr.V: The left- and right singular vectors of cross-TPDM, but transformed in positive reals with [trans](#)

---

to.alpha.2	<i>Probability integral transformation</i>
------------	--

---

**Description**

Performs transformation to make all of the margins follow a Frechet distribution with tail-index  $\alpha = 2$ .

**Usage**

```
to.alpha.2(data, orig = NULL)
```

**Arguments**

data	A $t \times n$ dimensional, numeric Data-matrix with t: Number of time steps and n: Number of grid points/stations
orig	If known: original distribution of data (currently implemented: 'normal' or 'gamma'), else: NULL

**Value**

Data-matrix of same dimension as 'data', but in Frechet-margins with tail-index 2

---

trans	<i>transformation function</i>
-------	--------------------------------

---

**Description**

Applies the transformation  $t(x) = \log(1 + \exp(x))$

**Usage**

```
trans(x)
```

**Arguments**

x	Real vector
---	-------------



**Details**

Transformation from real vector in real, positive vector under preservation of Frechet-distribution.

**Value**

Real, positive vector, containing the result of transformation function.

**Author(s)**

Yuing Jiang, Dan Cooley

**References**

Cooley & Thibaud (2019) <doi:10.1093/biomet/asz028>

**See Also**

[svd.tpdm](#), [pca.tpdm](#)

---

 wrapper.EPI

*Wrapper function*


---

**Description**

Handles all steps for estimation of EPI from raw-data: 1) Preprocessing into Frechet-Margins 2) Estimation of TPDM 3) Calculation of Principal Components 4) Estimation of EPI

**Usage**

```

wrapper.EPI(
  X,
  Y = NULL,
  q = 0.98,
  anz_cores = 1,
  clust = NULL,
  m = 1:10,
  thr_EPI = NULL
)

```

**Arguments**

X	A t x n dimensional Data-matrix with t: Number of time steps and n: Number of grid points/stations
Y	Optional: Same as X but for second variable: If Y!=NULL, cross-TPDM instead of TPDM and SVD instead of PCA is computed
q	Threshold for computation of TPDM. Only data above the 'q'-quantile will be used for estimation. Choose such that $0 < q < 1$ .

anz_cores	Number of cores for parallel computing (default: 5)
clust	Optional_ Uf clust = NULL, no declustering is performed. Else, declustering according to cluster-length 'clust'
m	Numeric vector: Containing the principal components/expansion coefficients (in case of Y!=NULL) from which the EPI shall be computed (default: modes = c(1:10), calculates the EPI on first ten principle Components)
thr_EPI	Only if Y!=NULL: Threshold for computation of TEPI. Expansion-coefficients that exceed the 'q'-quantile will be used for estimation. Choose such that $0 < q < 1$ .

### Value

In case of Y =NULL: A list containing:

- basis: The Eigenvectors of TPDM
- pc: The principal components of TPDM
- extremal.basis: The Eigenvectors of TPDM but transformed in positive reals with [trans](#)
- EPI: Extremal pattern index

In case of Y !=NULL: A list containing:

- U, V: The left- and right singular Vectors of cross-TPDM
- extr.U, extr.V: The left- and right singular vectors of cross-TPDM, but transformed in positive reals with [trans](#)
- pcU, pcV: The left- and right expansion coefficients of cross-TPDM
- EPI: Extremal pattern index
- TEPI: Threshold-based extremal pattern index

### References

Szemkus & Friederichs 2023

### Examples

```
data <- precipGER

result <- wrapper.EPI(data$pr, m = 1:50)

rbPal <- colorRampPalette(c('blue', 'white', 'red'))
Col <- rbPal(10)[as.numeric(cut(result$basis[,2],breaks = 10))]
plot(data$lat, data$lon,col=Col)
plot(data$date, result$EPI, type='l')
```

# Index

## \* datasets

precipGER, [7](#)

compute.EPI, [2](#)

decls, [3](#)

est.element.tpdm (est.tpdm), [4](#)

est.row.tpdm, [4](#)

est.row.tpdm (est.tpdm), [4](#)

est.tpdm, [3](#), [4](#), [4](#), [6](#), [7](#)

invTrans, [5](#)

pca.tpdm, [6](#), [6](#), [9](#)

precipGER, [7](#)

svd.tpdm, [6](#), [7](#), [9](#)

to.alpha.2, [8](#)

trans, [6](#), [8](#), [8](#), [10](#)

wrapper.EPI, [9](#)