



HAL
open science

The satdad R package: Version 1.0.0 Vignette

Cécile Mercadier

► **To cite this version:**

| Cécile Mercadier. The satdad R package: Version 1.0.0 Vignette. 2023. hal-04011079v2

HAL Id: hal-04011079

<https://hal.science/hal-04011079v2>

Submitted on 29 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The **satdad** R package: Version 1.1.0 Vignette

Cécile Mercadier

2023-03-29

1. Introduction & Installation

The name **satdad** is an acronym formed by the initials of **S**ensitivity **A**nalysis **T**ools for **D**ependence and **A**symptotic **D**ependence. The **satdad** R package provides tools for analyzing tail dependence in any sample or in particular theoretical models, namely **Mevlog** and **ArchimaxMevlog**. The package uses only theoretical and non parametric methods, without inference. Other tools and implementations will be added later to complete this first version. The primary goals of the package are to:

- Provide (a)symmetric multivariate extreme value models in any dimension **Mevlog** as some **Archimax** versions **ArchimaxMevlog**. *Let us emphasize `gen.ds`, which generates easily tail dependence structure.*
- Provide theoretical and empirical indices to order tail dependence. *Let us emphasize `tsic` and `tsicEmp` which compute and estimate **T**ail **S**uperset **I**mportance **C**oefficients.*
- Provide theoretical and empirical graphical methods to visualize tail dependence. *Let us emphasize the theoretical and empirical tail dependograph plotted by `graphs` and `graphsEmp`.*

The latest official release version can be obtained via

```
install.packages("satdad")
```

and loaded by

```
library(satdad)
```

The mentioned R packages are used as dependencies as following:

- **igraph** functions `graph`, `layout.fruchterman.reingold`, and `plot.igraph` are called.
- The `map` function is imported from `maps`.
- The `setparts` function is imported from `partitions`.
- The `danube` dataset is extracted from `graphicalExtremes` for the illustrative session.

2. Theoretical modelling

Consider $\mathbf{X} = (X^{(1)}, \dots, X^{(d)})$ a d -variate random vector. Under standard Frechet margins, a multivariate extreme value (mev) random vector \mathbf{X} has the cumulative distribution function

$$\mathbb{P}\left(X^{(1)} \leq x_1, \dots, X^{(d)} \leq x_d\right) = \exp\left(-\ell\left(\frac{1}{x_1}, \dots, \frac{1}{x_d}\right)\right),$$

for ℓ a stable tail dependence function. When restricting to (a)symmetric logistic dependence structure, such mev is called **Mevlog** in this package. It will also include mev with more general GEV margins.

Now, consider $\mathbf{U} = (U^{(1)}, \dots, U^{(d)})$ a d -variate random vector. Assume that the margins $U^{(t)}$, for $t = 1, \dots, d$, follow the standard uniform distribution. The cumulative distribution function of \mathbf{U} is then its copula function, $\mathbb{P}(U^{(1)} \leq x_1, \dots, U^{(d)} \leq x_d) = C(x_1, \dots, x_d)$. Assume that the copula function has the following form

$$C(x_1, \dots, x_d) = \psi(\ell(\psi^{-1}(x_1), \dots, \psi^{-1}(x_d)))$$

where ℓ is a stable tail dependence function and ψ is the generator of a d -variate Archimedean copula. One can refer to Charpentier et al. (2014) for the description of ψ and algorithms of simulation. Again, when ℓ is associated with a (a)symmetric logistic dependence structure, such model is called **ArchimaxMevlog** in this package.

2.1 Tail dependence structure

The tail structure of dependence is described by the stable tail dependence function ℓ defined above. In this package, it is saved in a **ds** object. The function **gen.ds** is used to generate these objects.

The multivariate asymmetric logistic model obtained through the option **type = "alog"**. It generates a multivariate asymmetric logistic model, which has been first introduced by Tawn (1990). We have

$$\ell(x_1, \dots, x_d) = \sum_{b \in B} \left(\sum_{i \in b} (\beta_{i,b} x_i)^{1/\alpha_b} \right)^{\alpha_b}$$

where B is the power set of $\{1, \dots, d\}$ (or a strict subset of the power set), the dependence parameters α_b lie in $(0, 1]$ and the collection of asymmetric weights $\beta_{i,b}$ are coefficients from $[0, 1]$ satisfying $\forall i \in \{1, \dots, d\}, \sum_{b \in B: i \in b} \beta_{i,b} = 1$. Missing asymmetric weights $\beta_{i,b}$ are assumed to be zero.

The class **ds** is a list that consists of:

- the dimension d .
- the type (**log** or **alog**).
- the list **sub** that corresponds to B . When **sub** is provided, the same list of subsets is returned, eventually sorted. When **sub = NULL** then **sub** is a subset of the power set of $\{1, \dots, d\}$. When the option **mns** is used, the latter integer indicates the cardinality of non singleton subsets in B .
- the dependence parameter **dep** = α or the vector of dependence parameters **dep** = $\{\alpha_b, b \in B\}$. When missing, these coefficients are obtained from independent standard uniform sampling.
- the list **asy** of asymmetric weights $\beta_{i,b}$ for $b \in B$ and $i \in b$. When missing, these coefficients are obtained from independent standard uniform sampling followed by a renormalization in order to satisfy the sum-to-one constraints.

Let us consider some examples.

```
## Construction of a ds object without using gen.ds
ds5 <- vector("list")
ds5$d <- 5
ds5$type <- "alog"
ds5$sub <- list(c(1,3),2:4,c(2,5))
ds5$asy <- list(c(1,.3),c(.5,1-.3,1), c(1-.5,1))
ds5$dep <- c(.2,.5,.3)
```

For larger dimensions, defining a **ds** object can become tricky, and the use of **gen.ds** is very helpful. For example, a 10-dimensional asymmetric tail dependence structure can be randomly created as follows.

```
## Three constructions of ds object by using gen.ds
# only d is given, sub, asy and dep are randomly sampled
ds10 <- gen.ds(d = 10)
```

```

# d and sub are given, asy and dep are randomly sampled
ds10 <- gen.ds(d = 10, sub = list(1:2,1:7,3:5,7:10))
# d is given, mnns indicates the cardinality of non singleton subsets in B
# sub, asy and dep are randomly sampled
ds10 <- gen.ds(d = 10, mnns = 4)

```

The symmetric case can be obtained through the `type = "log"` option in the `gen.ds` function, which yields a multivariate symmetric logistic model. This model is a well-known generalization of the bivariate extreme value logistic model introduced by Gumbel (1960). The parameter `dep` (with $0 < \text{dep} \leq 1$) is the only parameter needed to write the following equation

$$\ell(x_1, \dots, x_d) = \left(\sum_{i=1}^d x_i^{1/\text{dep}} \right)^{\text{dep}}.$$

If the parameter `dep` is missing, the function `gen.ds` will randomly generate its value from a standard uniform distribution.

For example, to obtain a 3-dimensional symmetric tail dependence structure with a randomly generated dependence parameter, you can use the following code

```

ds3 <- gen.ds(d = 3, type = "log")
ds3$dep

```

```
## [1] 0.7879912
```

If you know the value of the dependence parameter, you can specify it by setting `ds3$dep <- .3`, or by using the `dep` argument directly

```
ds3 <- gen.ds(d = 3, type = "log", dep = .3)
```

2.2 Sampling models

As mentioned at the beginning of the previous section, the `satdad` package studies both the `Mevlog` and `ArchimaxMevlog` theoretical models.

Samples of the MEV random vector with logistic dependence structures can be obtained via the `rMevlog` function using Algorithms 2.1 and 2.2 in Stephenson(2003).

```

n <- 1000
sample.frechet <- rMevlog(n, ds5) # standard Frechet margins
loc <- runif(5)
scale <- runif(5, 1, 2)
shape <- runif(5, -1, 1)
mar.gev <- cbind(loc, scale, shape)
sample.gev <- rMevlog(n, ds5, mar = mar.gev) # GEV margins all distinct
sample.samegev <- rMevlog(n, ds5, mar = c(-1,0.1,1)) # Gumbel margins

```

In addition, the package provides functions for computing the stable tail dependence function (`e11Mevlog`), cumulative distribution function (`pMevlog`), and probability density function (`dMevlog`) of the `Mevlog` distribution. The following are examples of commands for these functions, but without evaluation.

```

x5 <- runif(5)
e11Mevlog(x5, ds5)
pMevlog(x5, ds5) # cdf under standard Frechet margins
pMevlog(x5, ds5, mar = c(1,1,0)) # cdf under standard Gumbel margins
dMevlog(x5, ds5) # pdf under standard Frechet margins

```

In addition to multivariate extreme value logistic models, referred to as `Mevlog`, the `satdad` package provides some particular cases of `ArchimaxMevlog`. We follow here Algorithm 4.1 of p. 124 in Charpentier et al. (2014). Let ψ defined by $\psi(x) = \int_0^\infty \exp(-xt) dF_V(t)$, where F_V is the cumulative distribution function of a positive random variable.

We define the random vector (U_1, \dots, U_d) as $U_i = \psi(-\log(Y_i)/V)$ where

- Z has a multivariate extreme value distribution with stable tail dependence function ℓ ; here Z has standard Frechet margins,
- $(Y_1, \dots, Y_d) = (\exp(-1/Z_1), \dots, \exp(-1/Z_d))$ is the margin transform of \mathbf{Z} so that \mathbf{Y} is sampled from the extreme value copula associated with ℓ ,
- V has the distribution function F_V ,
- Y and V are independent.

Then, \mathbf{U} is sampled from the Archimax copula

$$C(x_1, \dots, x_d) = \psi(\ell(\psi^{-1}(x_1), \dots, \psi^{-1}(x_d))) .$$

The package provides `ArchimaxMevlog` realizations of random vectors \mathbf{U} . The cases covered by the `satdad` package are as follows:

– ψ is one among three types:

- $\psi(t) = \exp(-t)$; set `dist = "ext"`.
- $\psi(t) = \frac{\text{lambda}}{t + \text{lambda}}$; set `dist = "exp"` and `dist.param = lambda`.
- $\psi(t) = \frac{1}{(t + \text{scale})^{\text{shape}}}$; set `dist = "gamma"` and `dist.param = c(shape, scale)`.

– ℓ is the stable tail dependence function (stdf) associated with (a)symmetric logistic extreme value models.

`ArchimaxMevlog` samples are obtained via

```
n <- 1000
sample.ext <- rArchimaxMevlog(n, ds5, dist = "ext")
lambda <- runif(1, 1, 2)
sample.exp <- rArchimaxMevlog(n, ds5, dist = "exp", dist.param = lambda)
shape <- runif(1, 1, 2)
scale <- runif(1, 1, 2)
sample.gamma <- rArchimaxMevlog(n, ds5, dist = "gamma", dist.param = c(shape, scale))
```

The `satdad` package provides functions for computing ℓ , C , ψ , and ψ^{-1} for `ArchimaxMevlog` models. Specifically, `ellArchimaxMevlog`, `copArchimaxMevlog`, `psiArchimaxMevlog` and `psiinvArchimaxMevlog` can be used.

```
x <- runif(5)
ellMevlog(x, ds5)
```

```
## [1] 1.501714
```

```
ellArchimaxMevlog(x, ds5)
```

```
## [1] 1.501714
```

```
copArchimaxMevlog(x, ds5, dist = "ext")
```

```
## [1] 0.03070637
```

```
copArchimaxMevlog(x, ds5, dist = "exp", dist.param = lambda)
```

```
## [1] 0.1058089
```

```
copArchimaxMevlog(x, ds5, dist = "gamma", dist.param = c(shape, scale))
```

```
## [1] 0.109316
```

2.3 Measures and plots of the tail dependence

The tail dependence is completely characterized by the stdf ℓ , or equivalently by the `ds` object in this package. Summaries and graphical tools are obviously appreciated.

Well known extremal coefficients (*ec*), introduced by Tiago de Oliveira, J. (1962/63) and Smith (1990), are available in `satdad`. However, the focus is on the tail superset importance coefficients, which were introduced in Mercadier and Roustant (2019) and upper bounded in Mercadier and Ressel (2021). We believe that they also offer an interesting perspective on the description of the structure of the stable tail dependence function.

The theoretical functional decomposition of the variance of the stdf ℓ consists in writing

$$D(\ell) = \sum_{I \subseteq \{1, \dots, d\}} D_I(\ell)$$

where $D_I(\ell)$ measures the variance of $\ell_I(U_I)$ the term associated with subset I in the Hoeffding-Sobol decomposition of ℓ ; note that U_I represents a random vector with independent standard uniform entries. Fixing a subset of components I , the theoretical tail superset importance coefficient (*tsic*) is defined as

$$\Upsilon_I(\ell) = \sum_{J \supseteq I} D_J(\ell).$$

An integral representation of the superset importance coefficient is provided by Formula (9) of Liu and Owen (2006). See also Mercadier and Roustant (2019) for its use in the extreme value context. Thus, the *tsic* here is the value of $\Upsilon_I(\ell)$ obtained by Monte Carlo methods from the integral formula (3) in Mercadier and Roustant (2019).

```
res.tsic5 <- tsic(ds5)
```

```
as.character(res.tsic5$subsets)
```

```
## [1] "1:2" "c(1, 3)" "c(1, 4)" "c(1, 5)" "2:3" "c(2, 4)" "c(2, 5)" "3:4" "c(3, 5)" "4:5"
```

```
res.tsic5$tsic
```

```
## [1] 1.251700e-32 6.891844e-04 1.403926e-32 8.344669e-33 1.806057e-04 3.358261e-04 1.994653e-03 9.55
```

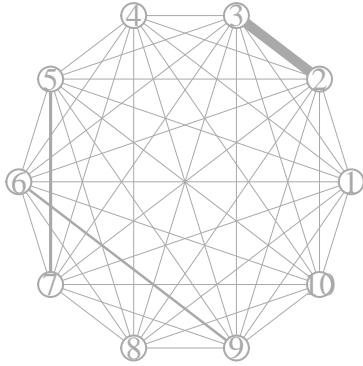
The `graphs` function in `satdad` implements the methodology introduced in Mercadier and Roustant (2019). The default option `which = taildependograph` draws the PAIRWISE *tsic* in a graphical representation called the tail dependograph. The command is as follows.

```
oldpar <- par(mfrow=c(1,2))
```

```
graphs(ds10) # (left) the nodes are plotted on an invisible circle
```

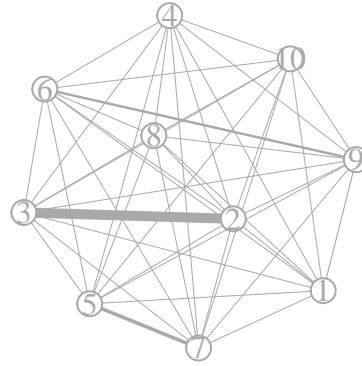
```
graphs(ds10, random = TRUE) # (right) the position of the nodes are random
```

Tail Dependograph



n.MC = 1000 thick.td = 5

Tail Dependograph

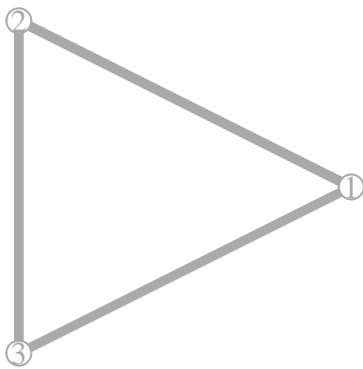


n.MC = 1000 thick.td = 5

```
par(oldpar)
```

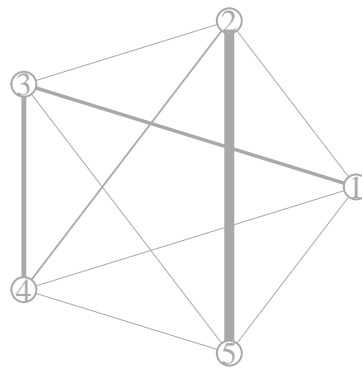
```
oldpar <- par(mfrow=c(1,2))
graphs(ds3) # (left) the symmetric structure
graphs(ds5) # (right) the asymmetric structure constructed "manually"
```

Tail Dependograph



n.MC = 1000 thick.td = 5

Tail Dependograph



n.MC = 1000 thick.td = 5

```
par(oldpar)
```

A theoretical upper bound for $\Upsilon_I(\ell)$ is given by Theorem 2 in Mercadier and Ressel (2021) which states that

$$\Upsilon_I(\ell) \leq \frac{2(|I|!)^2}{(2|I| + 2)!}$$

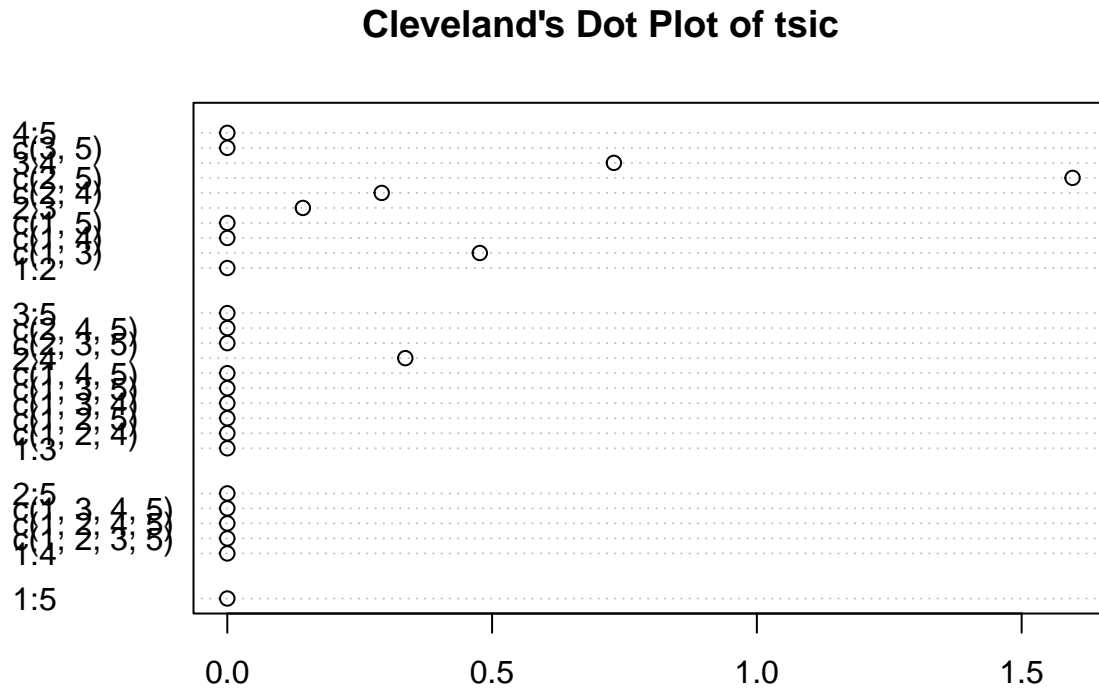
for any stdf ℓ . This allows for meaningful comparison of these indices, regardless of the cardinality of I , using

the expression

$$\frac{\Upsilon_I(\ell)}{D(\ell)} \times \frac{(2|I| + 2)!}{2(|I|!)^2}.$$

The option `sobol = TRUE` provides the renormalization by $D(\ell)$, while `norm = TRUE` multiplies by the inverse of the upper bound. The Cleveland dot plot is a useful tool to globally compare these coefficients.

```
plotClev(ds5)
```



The variance contribution $D_I(\ell)$ are referred to as the tail importance coefficient (tic) in this package, and should not be confused with the previously mentioned tsic, where “s” denotes supersets. The sobol version of tic, defined as

$$S_I(\ell) = \frac{D_I(\ell)}{D(\ell)}$$

can also be computed using this package.

```
res.tic5 <- tic(ds5, ind = "with.singletons", sobol = TRUE)
sobol5 <- res.tic5$tic # which sum should be 1
```

Well known extremal coefficients (ec) can be computed and visualized as follows.

```
res.ec10 <- ec(ds10)
as.character(res.ec10$subsets)
```

```
## [1] "1:2"      "c(1, 3)" "c(1, 4)" "c(1, 5)" "c(1, 6)" "c(1, 7)" "c(1, 8)" "c(1, 9)" "c(1, 10)"
## [13] "c(2, 6)" "c(2, 7)" "c(2, 8)" "c(2, 9)" "c(2, 10)" "3:4"     "c(3, 5)" "c(3, 6)" "c(3, 7)"
## [25] "4:5"     "c(4, 6)" "c(4, 7)" "c(4, 8)" "c(4, 9)" "c(4, 10)" "5:6"     "c(5, 7)" "c(5, 8)"
## [37] "c(6, 8)" "c(6, 9)" "c(6, 10)" "7:8"     "c(7, 9)" "c(7, 10)" "8:9"     "c(8, 10)" "9:10"
```

```
res.ec10$ec
```

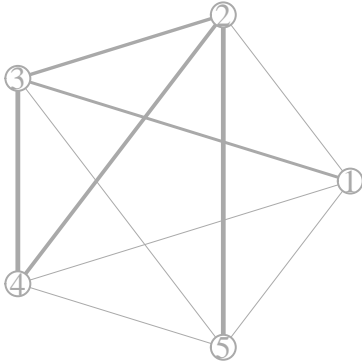
```
## [1] 1.839343 1.839343 1.492328 1.592642 1.170450 1.592930 1.431792 1.566809 1.357635 1.159015 1.542000
## [16] 1.998754 1.557315 1.542637 2.000000 2.000000 2.000000 1.574752 1.998754 1.557315 2.000000 1.624000
## [31] 1.539475 1.422518 2.000000 2.000000 1.526339 1.512494 1.547953 1.492949 1.478121 1.901416 2.000000
```

The option `which = "iecgraph"` in the `graphs` function of the package draws TWO minus the pairwise ec in a graphical representation.

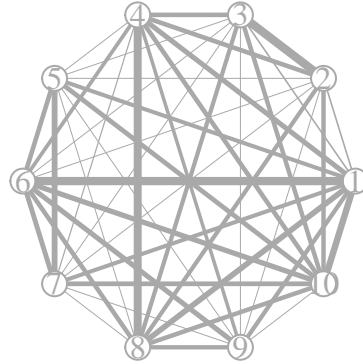

```
oldpar <- par(mfrow=c(1,2))
graphs(ds5, which = "iecggraph")
graphs(ds10, which = "iecggraph")
```

Inverse Extremal Coeff. Graph

Inverse Extremal Coeff. Graph



thick.ec = 5



thick.ec = 5

```
par(oldpar)
```

3. Empirical methods

The previous tail superset importance coefficients are computed by Monte Carlo approximation using the theoretical stable tail dependence function. When the latter is unknown, these indices are obtained from its non parametric estimation introduced by Huang (1992) in a bivariate setting and extended in de Haan and Resnick (1993).

Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be the **sample**, where each \mathbf{X}_s is a d -dimensional vector $X_s^{(t)}$ for $t = 1, \dots, d$.

Denote by n the sample size, and fix k as the threshold parameter.

Let $R_s^{(t)}$ denote the rank of $X_s^{(t)}$ among $X_1^{(t)}, \dots, X_n^{(t)}$, and set $\bar{R}_s^{(t)} = \min((n - R_s^{(t)} + 1)/k, 1)$.

3.1 Basics

Proposition 1 and Theorem 2 of Mercadier and Roustant (2019) indeed provide several rank-based expressions. Non parametric estimations of $\Upsilon_I(\ell)$, $D(\ell)$, $D_I(\ell)$, and $S_I(\ell)$ are as follows:

$$\hat{\Upsilon}_{I,k,n} = \frac{1}{k^2} \sum_{s=1}^n \sum_{s'=1}^n \prod_{t \in I} (\min(\bar{R}_s^{(t)}, \bar{R}_{s'}^{(t)}) - \bar{R}_s^{(t)} \bar{R}_{s'}^{(t)}) \prod_{t \notin I} \min(\bar{R}_s^{(t)}, \bar{R}_{s'}^{(t)})$$

$$\hat{D}_{k,n} = \frac{1}{k^2} \sum_{s=1}^n \sum_{s'=1}^n \prod_{t \in I} \min(\bar{R}_s^{(t)}, \bar{R}_{s'}^{(t)}) - \prod_{t \in I} \bar{R}_s^{(t)} \bar{R}_{s'}^{(t)}$$

$$\hat{D}_{I,k,n} = \frac{1}{k^2} \sum_{s=1}^n \sum_{s'=1}^n \prod_{t \in I} (\min(\bar{R}_s^{(t)}, \bar{R}_{s'}^{(t)}) - \bar{R}_s^{(t)} \bar{R}_{s'}^{(t)}) \prod_{t \notin I} \bar{R}_s^{(t)} \bar{R}_{s'}^{(t)}$$

and

$$\hat{S}_{I,k,n} = \frac{\hat{D}_{I,k,n}}{\hat{D}_{k,n}}.$$

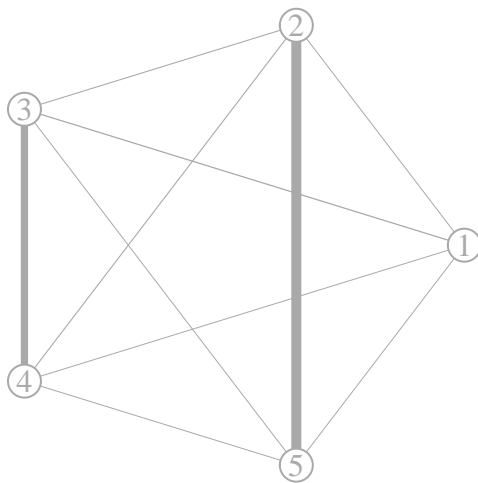
The functions `tsic`, `graphs`, `plotClev`, `ec` and `tic` have thus an empirical counterpart, namely, `tsicEmp`, `graphsEmp`, `plotClevEmp`, `ecEmp` and `ticEmp`. The `graphsEmp` function has another version called `graphsMapEmp` when coordinates of the nodes are provided.

```
res.ecEmp <- ecEmp(sample.ext, ind = "with.singletons", k = 100)
res.tsicEmp <- tsicEmp(sample.exp, ind = "all", k = 100)
res.ticEmp <- ticEmp(sample.gamma, ind = 4, k = 100)
```

The plots only are displayed.

```
graphsEmp(sample.ext, k = 100)
```

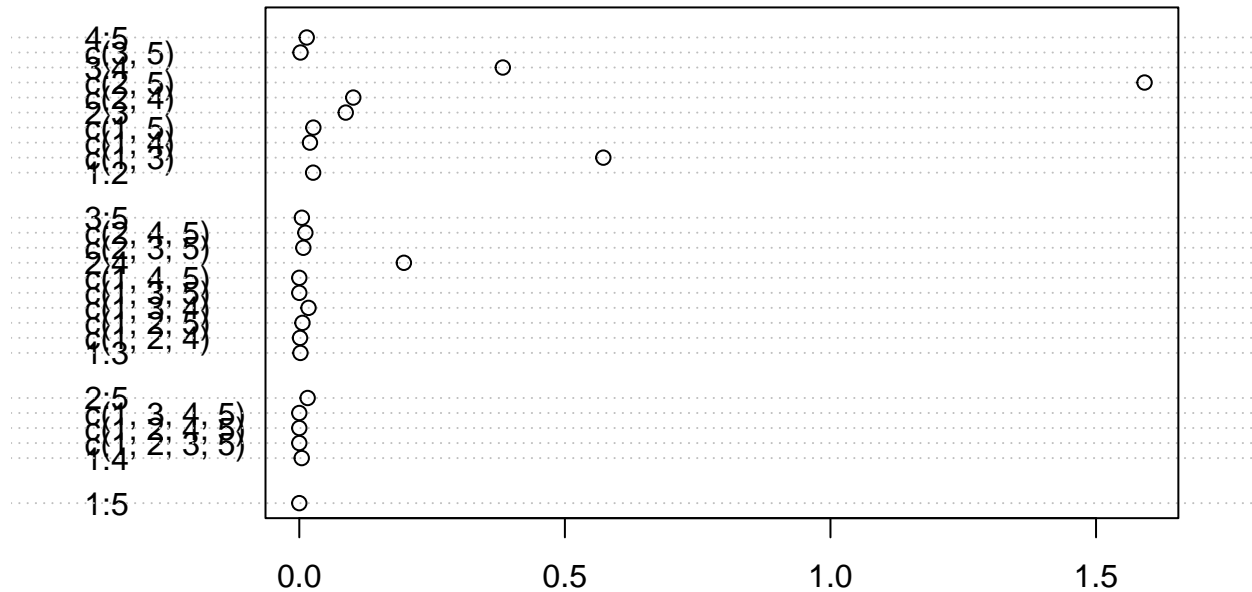
Emp. Tail Dependograph



k = 100 thick.td = 5

```
plotClevEmp(sample.exp, ind = "all", k = 100)
```

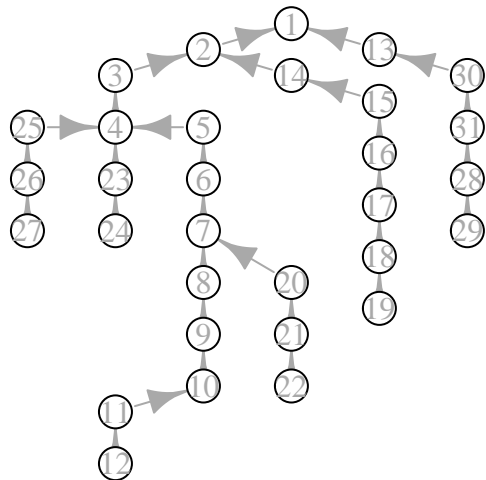
Cleveland's Dot Plot of Emp. tsic



3.2 The Danube dataset

The package `graphicalExtremes` implements the statistical methodology of Engelke and Hitz (2020), see also Asadi, Davison and Engelke (2015). The `danube` dataset in their package describes the river discharges for tributaries of the Danube.

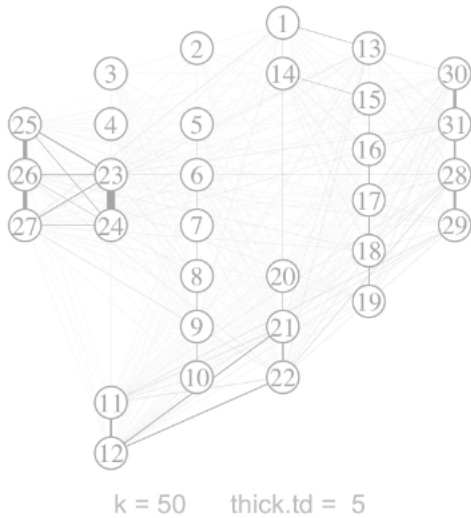
```
library(graphicalExtremes)
g <- igraph::graph_from_edgelist(danube$flow_edges)
loc <- as.matrix(danube$info[,c('PlotCoordX', 'PlotCoordY')])
plot(g, layout = loc, vertex.color = "white", vertex.label.color = "darkgrey")
```



The tail dependence understood through the global sensibility analysis is now provided.

```
dan <- danube$data_clustered
graphsEmp(dan, k=50, layout = loc)
```

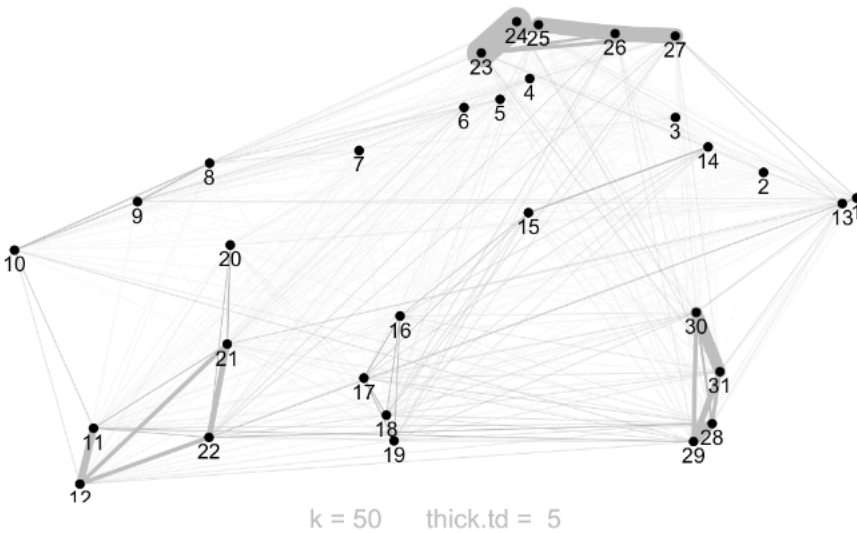
Emp. Tail Dependograph



The representation is also given on a realistic map.

```
lon <- as.numeric(unlist(danube$info[, "Long"]))
lat <- as.numeric(unlist(danube$info[, "Lat"]))*2
coord.dan <- list(lat = lat, lon = lon)
graphsMapEmp(dan, region = NULL, coord = coord.dan, k = 50, eps = 0.1)
```

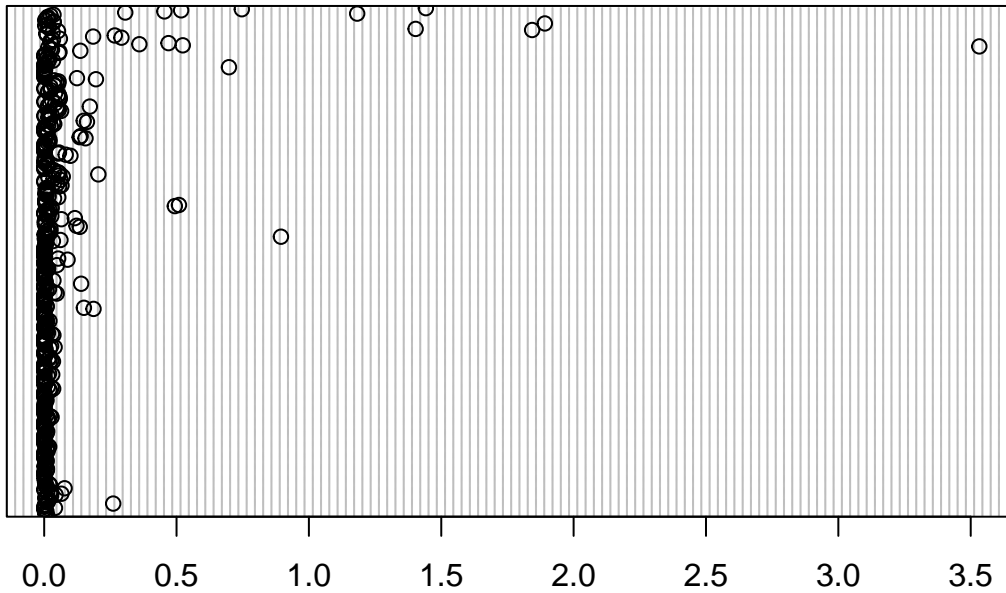
Emp. Tail Dependograph



A global comparison of the pairwise empirical tail superset importance coefficients is given by the empirical Cleveland's dot plot of the sample.

```
plotClevEmp(dan, k = 50, ind = 2, labels = FALSE)
```

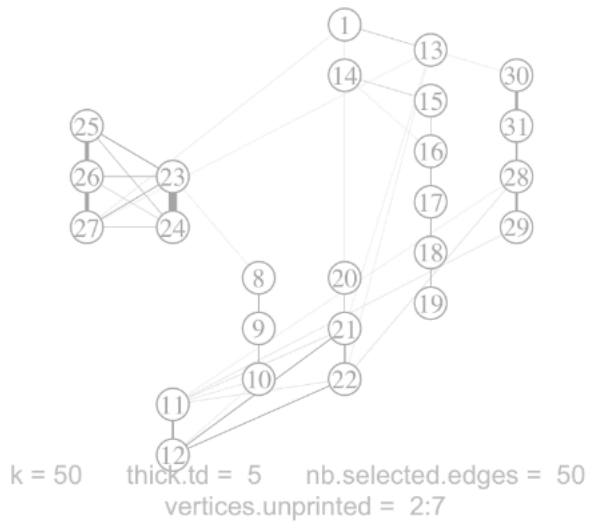
Cleveland's Dot Plot of Emp. tsic



Observing the largest points, we focus on the largest below.

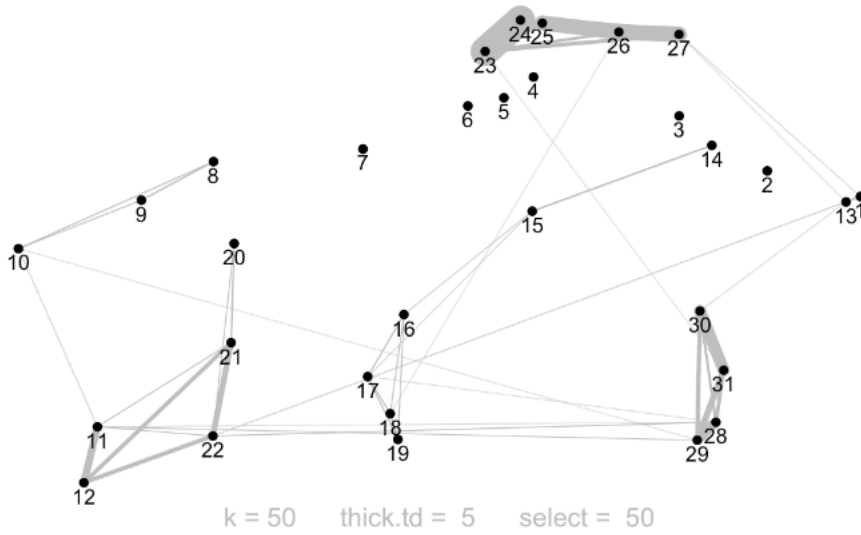
```
graphsEmp(dan, k=50, layout = loc, select = 50, simplify = TRUE)
```

Emp. Tail Dependograph



```
graphsMapEmp(dan, region = NULL, coord = coord.dan, k = 50, select = 50, eps = 0.1)
```

Emp. Tail Dependograph



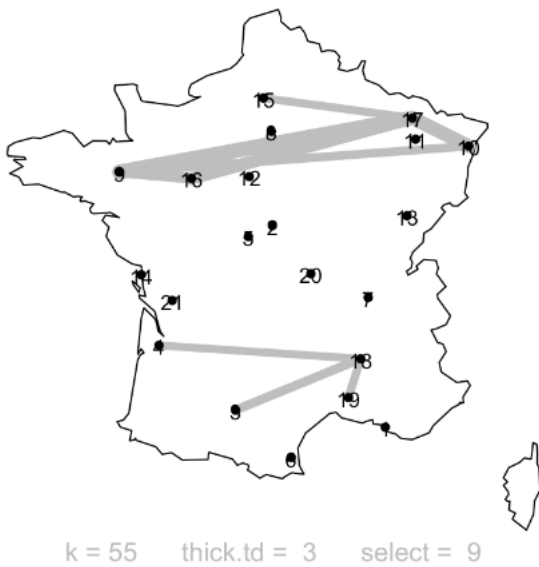
Graphs based on inverse extremal coefficients can also be obtained by adding the option `which = "iecgraph"`.

3.3 Other datasets

We provide below some figures from Mercadier and Roustant (2019), first about temperatures (France dataset) and then log returns then (Stock dataset).

```
## Figure 9 (a) of Mercadier and Roustant (2019).  
graphsMapEmp(sample = France$ymt, k = 55,  
              coord = France$coord, region = 'France', thick.td = 3, select = 9)
```

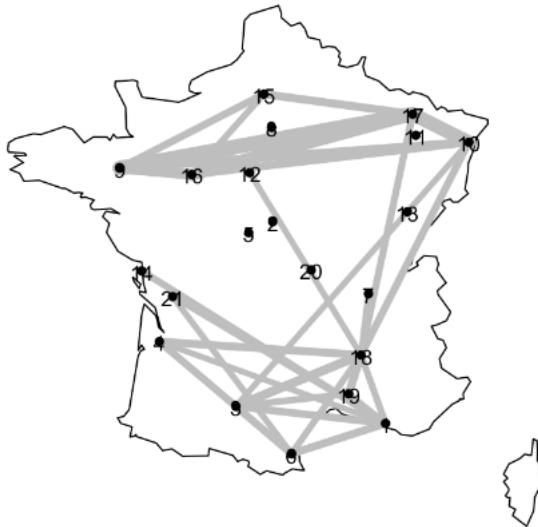
Emp. Tail Dependograph



```
## Figure 9 (b) of Mercadier and Roustant (2019).
```

```
graphsMapEmp(sample = France$ymt, k = 55,  
coord = France$coord, region = 'France', thick.td = 3, select = 30)
```

Emp. Tail Dependograph

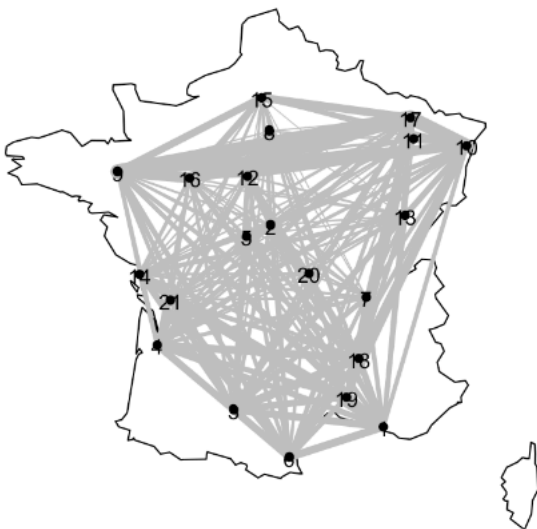


k = 55 thick.td = 3 select = 30

```
## Figure 9 (c) of Mercadier and Roustant (2019).
```

```
graphsMapEmp(sample = France$ymt, k = 55,  
coord = France$coord, region = 'France', thick.td = 3)
```

Emp. Tail Dependograph

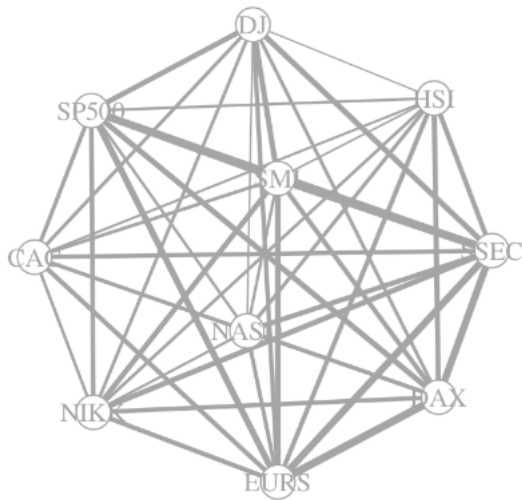


k = 55 thick.td = 3

```
## Figure 7(a) of Mercadier and Roustant (2019).
```

```
graphsEmp(Stock, k = 26, names = colnames(Stock), random = TRUE)
```

Emp. Tail Dependograph



k = 26 thick.td = 5

Figure 8(a) of Mercadier and Roustant (2019).

```
graphsEmp(Stock, k = 26, names = colnames(Stock), random = TRUE, select = 9)
```

Emp. Tail Dependograph

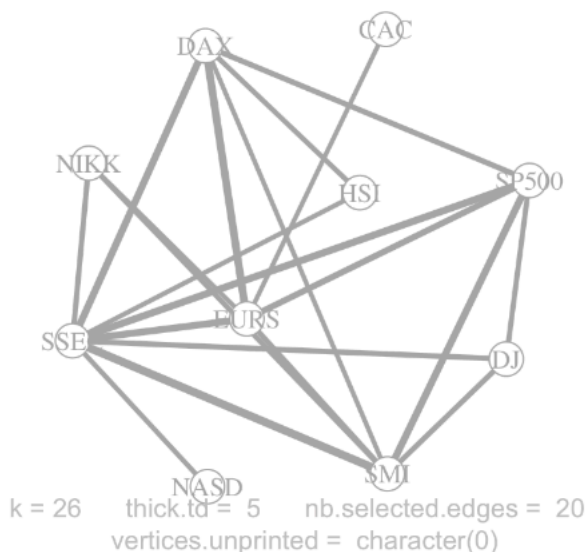


k = 26 thick.td = 5 nb.selected.edges = 9
vertices.unprinted = character(0)

Figure 8(b) of Mercadier and Roustant (2019).

```
graphsEmp(Stock, k = 26, names = colnames(Stock), random = TRUE, select = 20)
```


Emp. Tail Dependograph



4. Contents of the package `satdad`

More details on the package are given in the help pages associated with the following list of functions.

Method	Description
<code>copArchimaxMevlog</code>	cop-ell-psi-psiinv- functions for Archimax Mevlog models.
<code>dMevlog</code>	r-p-d-ell- functions for Mevlog models.
<code>ec</code>	Extremal coefficients for Mevlog models.
<code>ecEmp</code>	Empirical Extremal coefficients.
<code>ellArchimaxMevlog</code>	cop-ell-psi-psiinv- functions for Archimax Mevlog models.
<code>ellEmp</code>	Empirical stable tail dependence function.
<code>ellMevlog</code>	r-p-d-ell- functions for Mevlog models.
<code>France</code>	Dataset. Yearly Maxima of Temperature and coordinates of 21 French cities 1946-2000.
<code>gen.ds</code>	Generate and check a Mevlog tail dependence structure.
<code>graphs</code>	Graphs of the tail dependence structure for Mevlog models.
<code>graphsEmp</code>	Empirical graphs of the tail dependence structure.
<code>graphsMapEmp</code>	Empirical graphs drawn on geographical maps of the tail dependence structure.
<code>plotClev</code>	Cleveland's Dot Plots of the tail dependence structure.
<code>plotClevEmp</code>	Empirical Cleveland's Dot Plots of the tail dependence structure.
<code>pMevlog</code>	r-p-d-ell- functions for Mevlog models.
<code>psiArchimaxMevlog</code>	cop-ell-psi-psiinv- functions for Archimax Mevlog models.
<code>psiinvArchimaxMevlog</code>	cop-ell-psi-psiinv- functions for Archimax Mevlog models.
<code>rArchimaxMevlog</code>	r function for Archimax Mevlog models.
<code>rMevlog</code>	r-p-d-ell- functions for Mevlog models.
<code>Stock</code>	Dataset. Yearly maxima of Log Returns of ten stock indices 1990-2015.
<code>tic</code>	Tail importance coefficients for Mevlog models.

Method	Description
ticEmp	Empirical tail importance coefficients.
tsic	Tail superset importance coefficients for Mevlog models.
tsicEmp	Empirical tail superset importance coefficients.

If you have suggestions, or if you have encountered bugs, please contact me at mercadier@math.univ-lyon1.fr.

5. References

- Asadi, P., Davison, A.C. and Engelke, S. (2015). *Extremes on river networks*. The Annals of Applied Statistics, 9(4), 2023–2050.
- Becker, R. A., Wilks, A. R. (Original S code), Brownrigg, R. (R version), Minka, T. P. and Deckmyn A. (Enhancements). (2022) *maps : Draw Geographical Maps*. R package version 3.4.1.
- Charpentier, A., Fougères, A.-L., Genest, C. and Nešlehová, J.G. (2014) *Multivariate Archimax copulas*. Journal of Multivariate Analysis, 126, 118–136.
- Engelke, S. and Hitz, A.S. (2020). *Graphical models for extremes (with discussion)*. J. R. Stat. Soc. Ser. B Stat. Methodol., 82, 871–932.
- Fougères, A.-L., de Haan, L. and Mercadier, C. (2015). *Bias correction in multivariate extremes*. Annals of Statistics 43 (2), 903–934.
- Gumbel, E. J. (1960) *Distributions des valeurs extremes en plusieurs dimensions*. Publ. Inst. Statist. Univ. Paris, **9**, 171–173.
- de Haan, L. and Resnick, S. I. (1993). *Estimating the limit distribution of multivariate extremes*. *Communications in Statistics*. Stochastic Models 9, 275–309.
- Huang, X. (1992). *Statistics of bivariate extremes*. PhD Thesis, Erasmus University Rotterdam, Tinbergen Institute Research series No. 22.
- Liu, R. and Owen, A. B. (2006) *Estimating mean dimensionality of analysis of variance decompositions*. J. Amer. Statist. Assoc., **101**(474):712–721.
- Mercadier, C. and Ressel, P. (2021) *Hoeffding–Sobol decomposition of homogeneous co-survival functions: from Choquet representation to extreme value theory application*. Dependence Modeling, **9**(1), 179–198.
- Mercadier, C. and Roustant, O. (2019) *The tail dependograph*. Extremes, 22, 343–372.
- Smith, R. L. (1990) *Max-stable processes and spatial extremes*. Dept. of Math., Univ. of Surrey, Guildford GU2 5XH, England.
- Stephenson, A. (2002) *evd: Extreme Value Distributions*. R News, 2(2):31–32.
- Stephenson, A. (2003) *Simulating Multivariate Extreme Value Distributions of Logistic Type*. Extremes, **6**, 49–59.
- Tiago de Oliveira, J. (1962/63) *Structure theory of bivariate extremes, extensions*. Estudos de Matematica, Estatistica, e Economicos, 7:165–195.