

# Package ‘lcopula’

December 5, 2023

**Type** Package

**Title** Liouville Copulas

**Version** 1.0.7

**Description** Collections of functions allowing random number generations and estimation of 'Liouville' copulas, as described in Belzile and Neslehova (2017) <[doi:10.1016/j.jmva.2017.05.008](https://doi.org/10.1016/j.jmva.2017.05.008)>.

**License** GPL-3

**BugReports** <https://github.com/lbelzile/lcopula/issues>

**Imports** Rcpp (>= 0.11.4), utils, graphics, stats

**Depends** copula (>= 0.999-12), R (>= 2.12.0)

**Suggests** wdm

**LinkingTo** Rcpp

**RoxygenNote** 7.2.3

**Encoding** UTF-8

**LazyData** true

**NeedsCompilation** yes

**Author** Leo Belzile [aut, cre] (<<https://orcid.org/0000-0002-9135-014X>>),  
Christian Genest [aut, ctb],  
Alexander J. McNeil [ctb],  
Johanna G. Neslehova [ctb]

**Maintainer** Leo Belzile <[belzilel@gmail.com](mailto:belzilel@gmail.com)>

**Repository** CRAN

**Date/Publication** 2023-12-05 20:30:02 UTC

## R topics documented:

lcopula-package . . . . .	2
.ctspecdens . . . . .	3
.liouv.iTau_s . . . . .	4
.liouv.Tau_s . . . . .	4

.pickands.dir.uni . . . . .	5
.pickands.fun.uni . . . . .	5
.pliouv.opt_old . . . . .	6
airquality . . . . .	6
danube . . . . .	7
dirspecdens . . . . .	8
hmvevdiouv . . . . .	8
isliouv_m . . . . .	9
K.plot . . . . .	10
liouv.iTau . . . . .	11
liouv.maxim . . . . .	11
liouv.Tau . . . . .	12
Liouville . . . . .	13
Liouville_marginal . . . . .	14
negdirspecdens . . . . .	15
nutrient . . . . .	16
pickands.liouv . . . . .	17
pickands.plot . . . . .	18
pliouv.opt . . . . .	19
rarchi . . . . .	20
sliouv . . . . .	20
sliouv_m . . . . .	21
theta.bci . . . . .	22

<b>Index</b>	<b>24</b>
--------------	-----------

---

lcopula-package	<i>Liouville Copulas</i>
-----------------	--------------------------

---

## Description

**lcopula** provides functions to evaluate and sample Liouville copulas. The code is adapted from `Gumbelliouville.R` used in Genest and Neslehova (2013) and from the routines in `LiouvilleFunction.R` used by McNeil and Neslehova (2010) and coded by Alexander McNeil. The new implementation draws heavily on functions implemented in the `copula` package and derived from work by Hofert, Maechler and McNeil (2012).

*Warning:* a different definition is used by the latter and in the `copula` package for the Clayton copula. The implementation below uses the generator definition found in the first edition of the book *Quantitative Risk Management*.

The naming conventions are as follows: 'archi' and 'liouv' denote cores, prefixes 'd', 'p', 'r', 's', 'is' denote respectively density, distribution function, random number generation, survival function and inverse survival function. Suffixes are 'm' for marginal, '\_m' for simultaneous multiple margins and '\_p' for vectorized versions of the functions.

The functions mostly draws from the theory laid out in McNeil and Neslehova (2010) and Belzile (2014), which are the references for definitions and expressions.

The Archimedean families implemented are Clayton, Gumbel, Frank, Ali-Mikhail-Haq (abbreviated AMH) and Joe. Random number generation from the copulae is available via the function

`rliouv`. Method-of-moment estimators can be used for fitting for the Clayton and Gumbel families for bivariate samples, using the function `liouv.maxim.mm`. More general models can be fitted using `liouv.maxim`, but optimization may be slow. The function `theta.bci` allows to construct bootstrap confidence interval for the parameter  $\theta$ . Unless the method-of-moment estimator can be used, it is very slow since it relies on full optimization of each replicated sample under the model.

The copula and survival copula domains of attraction are implemented and the corresponding spectral density and Pickands dependence function can be computed and plotted in the bivariate case.

### Author(s)

Leo Belzile, using routines from Pr. Alexander J. McNeil, Pr. Johanna G. Neslehova. The `K.plot` was adapted from code provided by Pr. Christian Genest.

Maintainer: Leo Belzile <belzile@gmail.com>

### References

McNeil A.J. and Neslehova, J.G. (2010) From Archimedean to Liouville Copulas. *J. Multivar. Anal.*, **101**(8): 1772–1790.

Belzile L.R. (2014) Extremal and inferential properties of Liouville copulas. Master thesis, McGill.

Hofert, M., Maechler, M., and McNeil, A.J. (2012) Likelihood inference for Archimedean copulas in high dimensions under known margins. *J. Multivar. Anal.*, **110**, 133–150.

Genest, C. and Neslehova, J.G. (2013) Assessing and Modeling Asymmetry in Bivariate Continuous Data. In P. Jaworski, F. Durante, and W.K. Hardle (Eds.), *Copulae in Mathematical and Quantitative Finance*, Lecture Notes in Statistics, 91–114, Springer: Berlin Heidelberg.

Belzile, L.R. and Neslehova, J.G. (2017), Extremal attractors of Liouville copulas, *J. Multivar. Anal.*, **160**, 68–92.

### See Also

[copula](#) package

---

`.ctspecdens`

*Spectral density of Coles and Tawn extreme value distribution*

---

### Description

Spectral density of Coles and Tawn extreme value distribution

### Usage

```
.ctspecdens(param, dat, transform = TRUE)
```

### Arguments

<code>param</code>	vector containing the parameters $\alpha$ .
<code>dat</code>	matrix of pseudo-dat, of dimension $d$
<code>transform</code>	logical indicating whether parameters are on the log scale. Default to TRUE

**Value**

the value of the log likelihood for a sample of size  $n$

---

<code>.liouv.iTau_s</code>	<i>Moment estimate for theta derived from Kendall's <math>\tau</math></i>
----------------------------	---

---

**Description**

The moment estimates are based on inversion of the formula Kendall's  $\tau$  for Clayton or Gumbel Liouville copula

**Usage**

```
.liouv.iTau_s(tau_hat, family, alphavec)
```

**Arguments**

<code>tau_hat</code>	estimated Kendall's $\tau$ value from data
<code>family</code>	family of the Liouville copula. Either "clayton" or "gumbel"
<code>alphavec</code>	vector of Dirichlet allocations (must be a vector of integers)

**Value**

Value of theta

---

<code>.liouv.Tau_s</code>	<i>Computes Kendall's <math>\tau</math> for Clayton or Gumbel Liouville copula</i>
---------------------------	--

---

**Description**

The function computes Kendall's  $\tau$  for the given model, given alphavec

**Usage**

```
.liouv.Tau_s(theta, family, alphavec)
```

**Arguments**

<code>theta</code>	parameter of the corresponding Archimedean copula
<code>family</code>	family of the Liouville copula. Either "clayton" or "gumbel"
<code>alphavec</code>	vector of Dirichlet allocations (must be a vector of integers)

**Value**

value of  $\tau$

---

.pickands.dir.uni      *Pickands dependence function for the copula domain of attraction of Liouville survival copulas*

---

### Description

Pickands dependence function as in *Belzile (2014), Proposition 40 and Example 4* Returns the Pickands dependence function of the copula domain of attraction of the survival copula, the scaled Dirichlet extreme value model. Currently only implemented in the bivariate case. Setting  $\rho=1$  yields the same output as the function in the `evd` package.

### Usage

```
.pickands.dir.uni(t, alpha, rho)
```

### Arguments

t	pseudo-angle in (0,1)
alpha	vector of Dirichlet allocations. Currently must be of length 2
rho	index of regular variation parameter

### Value

value of Pickands function for the scaled extremal Dirichlet model

---

.pickands.fun.uni      *Pickands dependence function for the copula domain of attraction of Liouville copulas*

---

### Description

Pickands dependence function as in *Belzile (2014), Proposition 41* Returns the Pickands dependence function of the copula domain of attraction of the copula. This is only derived and implemented in the bivariate case.

### Usage

```
.pickands.fun.uni(t, rho = 0.5, alpha = c(1, 1))
```

### Arguments

t	pseudo-angle in (0,1)
rho	index of regular variation parameter
alpha	vector of Dirichlet allocations. Currently must be of length 2

**Value**

value of Pickands function

---

<code>.pliouv.opt_old</code>	<i>Old code for the copula likelihood function for Liouville copulas</i>
------------------------------	--

---

**Description**

The function is used internally for optimization.

**Usage**

```
.pliouv.opt_old(theta, data, family, alphavec, MC.approx = TRUE)
```

**Arguments**

<code>theta</code>	parameter of the corresponding Archimedean copula
<code>data</code>	sample matrix from a Liouville copula
<code>family</code>	family of the Liouville copula. Either "clayton", "gumbel", "frank", "AMH" or "joe"
<code>alphavec</code>	vector of Dirichlet allocations (must be a vector of integers)
<code>MC.approx</code>	whether to use Monte-Carlo approximation for the inverse survival function (default is TRUE)

**Value**

value of marginal density

---

<code>airquality</code>	<i>Daily air pollutant measures in Leeds</i>
-------------------------	--

---

**Description**

The `airquality` data frame consists of measurements of pollutants in the city of Leeds. In the saying of Boldi and Davison (2007): "The dataset comprise daily series of monitoring measurements of ozone levels (O3), nitrogen dioxide (NO2), nitrogen oxide (NO) and particulate matter (PM10), in the city centre of Leeds, UK, over 1994–1998. Levels of the gases are measured in parts per billion, and those of PM 10 in micrograms per cubic metre. "

**Details**

Downloaded from <https://uk-air.defra.gov.uk/>. (c) Crown 2015 copyright Defra via uk-air.defra.gov.uk, licenced under the Open Government Licence (OGL).

## References

- Heffernan, J., Tawn, J. (2004) A conditional approach for multivariate extreme values (with discussion). *J. R. Stat. Soc., Ser. B Stat. Methodol.* **66**(3), 497–546.
- Sabourin, A. , Naveau, P., Fougères, A.-L. (2013) Bayesian Model averaging for Multivariate extremes. *Extremes*, **16**(3), 325–350.
- Boldi, M.O., Davison, A.C. (2007) A mixture model for multivariate extremes. *J. R. Stat. Soc., Ser. B Stat. Methodol.* **69**(2), 217–229.
- Cooley, D., Davis, R., Naveau, P. (2010) The pairwise beta distribution: A flexible parametric multivariate model for extremes. *J. Multivar. Anal.* **101**(9), 2103–2117.

---

danube

*Ranks of preprocessed monthly Danube river flow measurements*

---

## Description

The danube dataset contains ranks of base flow observations from the Global River Discharge project of the Oak Ridge National Laboratory Distributed Active Archive Center (ORNL DAAC), a NASA data center. The measurements are monthly average flow rate for two stations situated at Scharding (Austria) on the Inn river and at Nagymaros (Hungary) on the Danube. The data have been pre-processed to remove any time trend. Specifically, Bacigal et al. (2011) extracted the raw data, and obtain the fast Fourier transformed centered observations. The negative spectrum is retained and a linear time series model with 12 seasonal components is fitted. Residuals are then extracted and AR model fitted to the series, the selection being done based on the AIC criterion with imposed maximum order of 3 and the number of autoregressive components may differ for each series.

## Format

This data frame contains the following columns:

- inn** A numeric vector containing the rank of prewhitened level observations of the Inn river at Nagymaros.
- donau** A numeric vector containing the rank of prewhitened level observations of the Donau river at Scharding.

## Source

Vorosmarty, C.J., Fekete, B.M., Tucker, B.A. (1998). Global River Discharge, 1807–1991, V. 1.1(RivDIS). Data set. Available from <http://doi.org/10.3334/ORNLDAAC/199>

## References

- Bacigal, T., JAGR, V., MesiAR, R. (2011) Non-exchangeable random variables, Archimax copulas and their fitting to real data. *Kybernetika*, **47**(4), 519–531.
- Genest, C. and Neslehova, J. G. (2013) Assessing and Modeling Asymmetry in Bivariate Continuous Data. In P. Jaworski, F. Durante, and W. K. Hardle (Eds.), *Copulae in Mathematical and Quantitative Finance*, Lecture Notes in Statistics, 91–114, Springer: Berlin Heidelberg.

---

dirspecdens	<i>Spectral density of scaled Dirichlet and negative Dirichlet extreme value distributions</i>
-------------	--

---

**Description**

Spectral density of scaled Dirichlet and negative Dirichlet extreme value distributions

**Usage**

```
dirspecdens(param, dat, d, transform = TRUE)
```

**Arguments**

param	vector containing the parameters $\alpha$ and $\rho$ , in this order.
dat	matrix of Frechet or Pareto observations, of dimension $n$ by $d$ .
d	dimension of model
transform	logical indicating whether parameters are on the log scale. Default to TRUE

**Details**

The function is provided as a wrapper and takes parameters on the log scale for  $\alpha$  ( $\rho$ ).

**Value**

the log-likelihood for the  $n$  sample

---

hmvevdlouv	<i>Spectral density of the CDA of survival copula and copula of Liouville vectors</i>
------------	---

---

**Description**

Computes the Liouville EV model or the scaled Dirichlet EV model spectral density

**Usage**

```
hmvevdlouv(w, alpha, rho, CDA = c("C", "S"), logdensity = FALSE)
```

**Arguments**

w	matrix of points in the unit simplex at which to evaluate the density
alpha	vector of Dirichlet allocations (strictly positive).
rho	parameter of limiting model corresponding to index of regular variation
CDA	copula domain of attraction of either the Liouville copula, C, or its survival copula S
logdensity	logical; whether to return the log density or not



**Value**

a vector with the same number of rows as w.

**Examples**

```
hmvevdliouv(seq(0.01,0.99,by=0.01), alpha=c(1,2), rho=0.2, CDA="C")
hmvevdliouv(seq(0.01,0.99,by=0.01), alpha=c(0.1,2), rho=0.2, CDA="S")
hmvevdliouv(seq(0.01,0.99,by=0.01), alpha=c(1,2), rho=0.2, CDA="S")
```

---

 isliouv\_m

---

*Multiple marginal inverse survival function of Liouville vectors*


---

**Description**

This function is a wrapper around `isliouvm`; it allows the user to treat all the data matrix simultaneously by applying different parameters to each margin.

**Usage**

```
isliouv_m(u, family, alphavec, theta)
```

**Arguments**

u	vector of survival probabilities
family	family of the Liouville copula. Either "clayton", "gumbel", "frank", "AMH" or "joe"
alphavec	vector of Dirichlet allocations (must be a vector of integers)
theta	parameter of the corresponding Archimedean copula

**Value**

a vector of same length as u with the quantile at 1-u

**Examples**

```
u <- rliouv(n = 10, family = "clayton", alphavec <- c(2,3), theta = 2)
isliouv_m(u=u, family="clayton", alphavec=c(2,3), theta=2)
```

---

K.plot *Kendall plot*

---

### Description

This function plots the expectation of the order statistics under the null hypothesis of independence against the ordered empirical copula values. The data is transformed to ranks.

### Usage

```
K.plot(data, add = F, ...)
```

### Arguments

data	a n by d matrix of observations
add	whether to superimpose lines to an existing graph. Default to F
...	additional arguments passed to points

### Details

The function uses `integrate` and may fail for large  $d$  or large  $n$ . If  $n > 200$ , the fallback is to generate a corresponding sample of uniform variates and to compare the empirical copula of the sample generated under the null hypothesis with the one obtained from the sample.

### Value

The Kendall plot corresponding to the data at hand

### Author(s)

Pr. Christian Genest (the code was adapted for the multivariate case)

### References

Genest & Boies (2003). Detecting Dependence with Kendall Plots, *The American Statistician*, **57**(4), 275–284.

### Examples

```
#Independence
K.plot(matrix(runif(2000),ncol=2))
#Negative dependence
K.plot(rCopula(n=1000,claytonCopula(param=-0.5,dim=2)),add=TRUE,col=2)
#Perfect negative dependence
K.plot(rCopula(n=1000,claytonCopula(param=-1,dim=2)),add=TRUE,col=6)
#Positive dependence
K.plot(rCopula(n=1000,claytonCopula(param=iTau(claytonCopula(0.3),0.5),dim=2)),add=TRUE,col=3)
#Perfect positive dependence
K.plot(rCopula(n=1000,claytonCopula(param=iTau(claytonCopula(0.3),1),dim=2)),add=TRUE,col=4)
```

---

liouv.iTau	<i>Moment estimate for theta derived from Kendall's tau formula</i>
------------	---

---

**Description**

The moment estimates are based on inversion of the formula Kendall's tau for Clayton or Gumbel Liouville copula

**Usage**

```
liouv.iTau(tau_hat, family, alphavec)
```

**Arguments**

tau_hat	estimated vector of Kendall's tau values
family	family of the Liouville copula. Either "clayton" or "gumbel"
alphavec	vector of Dirichlet allocations (must be a vector of integers)

**Value**

Vector of theta

**Examples**

```
liouv.iTau(0.5, family="gumbel", c(1,2))
liouv.iTau(0.5, family="clayton", c(3,2))
```

---

liouv.maxim	<i>Maximization of Liouville copula likelihood function</i>
-------------	---

---

**Description**

Two methods, either numerical optimization or method-of-moments

**Usage**

```
liouv.maxim(
  data,
  family,
  interval,
  boundary = NULL,
  lattice.mat = NULL,
  return_all = FALSE,
  MC.approx = TRUE
)
```

**Arguments**

data	sample matrix from a Liouville copula
family	family of the Liouville copula. Either "clayton", "gumbel", "frank", "AMH" or "joe"
interval	interval over which to look for theta (bounds for Nelder-Mead)
boundary	vector of endpoints for search of Dirichlet allocation parameters. Either boundary or lattice.mat can be supplied
lattice.mat	matrix of tuples of Dirichlet allocation parameters at which to evaluate the likelihood
return_all	should all results (as list) or only maximum value be returned. Defaults to FALSE
MC.approx	whether to use Monte-Carlo approximation for the inverse survival function (default is TRUE)

**Details**

A wrapper to `optim` using the Nelder-Mead algorithm or using the methods of moments, to maximize pointwise given every `alphavec` over a grid. Returns the maximum for `alphavec` and `theta`.

**Value**

a list with values of `theta` and Dirichlet parameter along with maximum found. Gives index of maximum amongst models fitted.

**Examples**

```
## Not run:
data <- rliouv(n=100, family="joe", alphavec=c(1,2), theta=2)
liouv.maxim(data=data, family="j", interval=c(1.25,3), boundary=c(2,2),return_all=TRUE)
lattice.mat <- t(combn(1:3,2))
liouv.maxim(data=data, family="j", interval=c(1.25,3), lattice.mat=lattice.mat, return_all=FALSE)
#data <- rliouv(n=1000, family="gumbel", alphavec=c(1,2), theta=2)
liouv.maxim.mm(data=data, family="gumbel", boundary=c(3,3),return_all=TRUE)
lattice.mat <- t(combn(1:3,2))
liouv.maxim.mm(data=data, family="gumbel", lattice.mat=lattice.mat, return_all=FALSE)

## End(Not run)
```

---

liouv.Tau

*Computes Kendall's tau for Clayton or Gumbel Liouville copula*


---

**Description**

The function computes Kendall's  $\tau$  for the given model, given `alphavec`

**Usage**

```
liouv.Tau(theta, family, alphavec)
```

**Arguments**

theta	parameter of the corresponding Archimedean copula
family	family of the Liouville copula. Either "clayton" or "gumbel"
alphavec	vector of Dirichlet allocations (must be a vector of integers)

**Value**

vector of  $\tau$

**Examples**

```
liouv.Tau(theta=2, family="gumbel", alphavec=c(1,2))
liouv.Tau(theta=1, family="clayton", alphavec=c(2,1))
```

---

Liouville	<i>Liouville copulas</i>
-----------	--------------------------

---

**Description**

Multivariate density, survival copula and random generation for the Liouville copulas.

**Usage**

```
rliouv(n = 100, family, alphavec, theta, reverse = FALSE)
pliouv(x, theta, family, alphavec)
dliouv(x, family, alphavec, theta, is.log = FALSE)
```

**Arguments**

n	sample size
family	family of the Liouville copula. Either "clayton", "gumbel", "frank", "AMH" or "joe"
alphavec	vector of Dirichlet allocations (must be a vector of integers) Specifies (implicitly) the dimension of sample
theta	parameter of the corresponding Archimedean copula
reverse	if TRUE, return sample from the corresponding survival copula
x	matrix of quantiles from a Liouville copula
is.log	if TRUE, will return the log-likelihood value

**Details**

`rliouv` generates draws from the Liouville copula. `dliouv` evaluates the density of an  $n$  by  $d$  matrix of observations. `pliouv` is the (survival) copula associated with the Liouville vector and is as such the multivariate distribution function for uniform observations.

Liouville copulas were introduced in McNeil and Neslehova (2010), generalizing Archimedean copulas. Like the latter, they are survival copulas, which means that the copula is evaluated using the (multivariate) survival function of Liouville vectors. See also `sliouv` for the latter.

The Liouville copula is by definition a survival copula. The function thus maps marginally observations from the unit interval to the positive half-line using the marginal inverse survival function `isliouvm` of the Liouville vector, and then evaluating the survival distribution at the resulting Liouville vector.

**Value**

either a matrix of dimension  $n$  by  $\text{length}(\text{alphavec})$  with the corresponding quantile, probability, survival probability or sample from the Liouville vector

**References**

McNeil A.J. and Neslehova, J.G. (2010) From Archimedean to Liouville Copulas. *J. Multivar. Anal.*, **101**(8): 1772–1790.

**See Also**

[Liouville\\_marginal](#)

**Examples**

```
## Not run:
##Multivariate density of Clayton Liouville copula
x <- rliouv(n = 100, family = "clayton", alphavec <- c(2,3), theta = 2)
dliouv(x=x, family="clayton", alphavec=c(2,3), theta=2, TRUE)
##Distribution function, multivariate sample
x <- rliouv(n=100, family="frank", theta=1.5, alphavec=c(2,3))
pliouv(theta=1.5, x=x, family="frank", alphavec=c(2,3))

## End(Not run)
```

---

Liouville\_marginal      *Liouville vectors marginal functions*

---

**Description**

Marginal density, distribution, survival and inverse survival functions for Liouville copulas or Liouville vectors. The inverse survival function of Liouville vectors is not available in closed-form and is obtained numerically by root-finding. As such, Monte-Carlo approximation have been considered for dealing with inference to avoid computational bottlenecks. Note: the arguments of `sliouv` are reversed since they are meant to be called inside `optim`. The functions borrow *psi* functions and their derivatives from the [copula-package](#).

**Usage**

```
sliouvm(x, family, alpha, theta)

pliouvm(x, family, alpha, theta)

isliouvm(u, family, alpha, theta)

dliouvm(x, family, alpha, theta)
```

**Arguments**

x	vector of quantiles from a Liouville copula (or a Liouville vector for the survival function , with support on the positive real line)
family	family of the Liouville copula. Either "clayton", "gumbel", "frank", "AMH" or "joe"
alpha	integer Dirichlet parameter
theta	parameter of the corresponding Archimedean copula
u	vector of quantiles or survival probabilities, (pseudo)-uniform variates

**Value**

a vector with the corresponding quantile, probability, survival probabilities

**Examples**

```
## Not run:
#Marginal density
samp <- rliouv(n = 100, family = "clayton", alphavec <- c(2,3), theta = 2)
dliouvm(x=samp[,1], family="clayton", alpha=2, theta=2)
sum(log(dliouvm(x=samp[,1], family="clayton", alpha=2, theta=2)))
#Marginal distribution and (inverse) survival function
x <- rliouv(n = 100, family = "gumbel", alphavec <- c(2,3), theta = 2)
pliouvm(x[,1], family="gumbel", alpha=alphavec[1], theta=2)
su <- sliouvm(1-x[,1], family="gumbel", alpha=alphavec[1], theta=2)
isliouvm(u=su, family="clayton", alpha=2, theta=2)
#pliouv is the same as sliouv(isliouvm)

## End(Not run)
```

---

negdirspecdens

*Spectral density of scaled negative Dirichlet extreme value distribution*


---

**Description**

Spectral density of scaled negative Dirichlet extreme value distribution

**Usage**

```
negdirspecdens(param, dat, d, transform = TRUE)
```

**Arguments**

**param** vector containing the parameters  $\alpha$  and  $\rho$ , in this order.  
**dat** matrix of Frechet or Pareto observations, of dimension  $n$  by  $d$ .  
**d** dimension of model  
**transform** logical indicating whether parameters are on the log scale. Default to TRUE

**Details**

The function is provided as a wrapper and takes parameters on the log scale for  $\alpha$  and  $\rho$ .

**Value**

the log-likelihood for the  $n$  sample

---

nutrient	<i>Women daily nutrient intake</i>
----------	------------------------------------

---

**Description**

The nutrient data frame consists of quintuples consisting of four day measurements for intake of calcium, iron, protein, vitamin A and C from women aged 25 to 50 in the United States as part of the "Continuing Survey of Food Intakes of Individuals" program. The processed data has 737 measurements from a cohort study of the United States Department of Agriculture (USDA) and it is available online from the University of Pennsylvania repository.

**Format**

This data frame contains the following columns:

**id** A numeric vector containing the identification of the participant.

**calcium** A numeric vector containing measured calcium.

**iron** A numeric vector containing iron measurements.

**protein** A numeric vector containing protein measurements.

**vitamin.a** A numeric vector containing vitamin A measurements.

**vitamin.c** A numeric vector containing vitamin C measurements.

**Source**

The survey data was processed by Dr. Andrew Wiesner for a course on multivariate statistics at The Pennsylvania State University.



## References

- Genest, C., Neslehova, J., Quessy, J.-F. (2012) Tests of symmetry for bivariate copulas. *Annals of the Institute of Statistical Mathematics*, **64**(4), 811–834.
- Genest, C. and Neslehova, J. G. (2013) Assessing and Modeling Asymmetry in Bivariate Continuous Data. In P. Jaworski, F. Durante, & W. K. Hardle (Eds.), *Copulae in Mathematical and Quantitative Finance*, Lecture Notes in Statistics, 91–114, Springer: Berlin Heidelberg.
- Li, B., Genton, M. G. (2013) Nonparametric identification of copula structures. *Journal of the American Statistical Association*, **108**(502), 666–675.

---

pickands.liouv	<i>Pickands dependence function for the copula domain of attraction of Liouville survival copulas</i>
----------------	---

---

## Description

Pickands dependence function as in *Belzile (2014), Proposition 40 and Example 4* and *Belzile (2014), Proposition 41*, assuming that the parameter alpha is integer-valued. Returns the Pickands dependence function of the copula domain of attraction (CDA) of the survival copula, the scaled Dirichlet extreme value model, or the CDA of the copula, the Liouville EV model.

## Usage

```
pickands.liouv(t, rho = 0.5, alpha = c(1, 1), CDA = c("C", "S"))
```

## Arguments

t	pseudo-angle in (0,1)
rho	index of regular variation parameter
alpha	vector of Dirichlet allocations. Currently must be of length 2
CDA	select the extremal attractor of the copula (C) or the survival copula (S)

## Value

value of Pickands function for the scaled Dirichlet EV model

## Examples

```
pickands.liouv(seq(0,1,by=0.01),1,c(0.1,0.3),CDA="S")
pickands.liouv(t = seq(0,1,by=0.01), rho = 0.5, alpha = c(1,3), CDA="C")
```

---

pickands.plot

*Plot Pickands dependence function for CDA of Liouville copulas*


---

### Description

The function will draw the Pickands dependence function for output in tikz if the corresponding function is selected.

### Usage

```
pickands.plot(
  rho,
  alpha,
  plot.new = TRUE,
  CDA = c("C", "S"),
  tikz = FALSE,
  ...
)
```

### Arguments

rho	index of regular variation parameter
alpha	vector of Dirichlet allocations. Currently must be of length 2
plot.new	boolean indicating whether a new plotting device should be called
CDA	whether to plot Pickands function for the extremal model of the copula (C) or the survival copula (S), which is the scaled Dirichlet
tikz	boolean specifying whether to prepare plot for tikz output. Defaults to F
...	additional arguments passed to lines

### Value

a plot of the Pickands dependence function

### Examples

```
pickands.plot(
  rho = 0.9,
  alpha = c(1,1),
  col = "slateblue1",
  CDA = "C")
pickands.plot(
  rho = 0.9,
  alpha = c(2,3),
  col = "slateblue2",
  CDA = "C",
  plot.new = FALSE)
pickands.plot(
```

```

rho = 0.5,
alpha = c(2,3),
col = "slateblue3",
CDA = "C",
plot.new = FALSE)
#Parameters for the Pickands function of the scaled Dirichlet need not be integer
pickands.plot(
  rho = 0.9,
  alpha = c(1,1),
  CDA = "S")
pickands.plot(
  rho = 0.9,
  alpha = c(0.2,0.5),
  col = "darkred",
  CDA = "S",
  plot.new = FALSE)
pickands.plot(
  rho = 0.8,
  alpha = c(1.2,0.1),
  col = "red",
  CDA = "S",
  plot.new = FALSE)

```

---

 pliouv.opt

---

*Internal code for the copula likelihood function of Liouville copulas*


---

## Description

The function is used internally for optimization.

## Usage

```
pliouv.opt(theta, data, family, alphavec, MC.approx = TRUE)
```

## Arguments

theta	parameter of the corresponding Archimedean copula
data	sample matrix from a Liouville copula
family	family of the Liouville copula. Either "clayton", "gumbel", "frank", "AMH" or "joe"
alphavec	vector of Dirichlet allocations (must be a vector of integers)
MC.approx	whether to use Monte-Carlo approximation for the inverse survival function (default is TRUE)

## Value

value of marginal density

---

rarchi                      *Archimedean copula sampler*

---

### Description

Sampler based on the Marshall-Olkin algorithm

### Usage

```
rarchi(n, family, d, theta)
```

### Arguments

n	sample size
family	family of the Archimedean copula. Either "clayton", "gumbel", "frank", "AMH" or "joe"
d	dimension of sample
theta	parameter of the Archimedean copula

### Value

a sample of dimension n by d from the Archimedean copula

### Examples

```
#Sample from a Gumbel Archimedean copula
rarchi(n = 100, "gumbel", d = 4, theta = 2)
#Sample from the independence copula
rarchi(n = 100, "gumbel", d = 4, theta = 1)
```

---

sliouv                      *Joint survival function of Liouville vectors*

---

### Description

sliouv returns the survival function of a Liouville vector. For the survival copula and the associated probability, see [pliouv](#).

### Usage

```
sliouv(theta, x, family, alphavec)
```

**Arguments**

theta	parameter of the corresponding Archimedean copula
x	an n by d matrix of observations, each on the positive real line
family	family of the Liouville copula. Either "clayton", "gumbel", "frank", "AMH" or "joe"
alphavec	d-vector of Dirichlet allocations (must be a vector of integers)

sliouv\_m

*Multiple marginal survival function for Liouville vectors***Description**

This function is a wrapper around `sliouv`; it allows the user to treat all the data matrix simultaneously by applying different parameters to each margin.

**Usage**

```
sliouv_m(x, family, alphavec, theta)
```

**Arguments**

x	sample from copula
family	family of the Liouville copula. Either "clayton", "gumbel", "frank", "AMH" or "joe"
alphavec	vector of Dirichlet allocations (must be a vector of integers)
theta	parameter of the corresponding Archimedean copula

**Value**

a matrix of same length as x with the survival probabilities

**Examples**

```
x <- rliouv(n = 100, family = "gumbel", alphavec <- c(2,3), theta = 2)
sliouv_m(x, family="gumbel", alphavec=c(2,3), theta=2)
all(sliouv_m(x, family="gumbel", alphavec=c(2,3), theta=2)[,1]-
  sliouv(x[,1], family="gumbel", alpha=2, theta=2)==0)
```

---

theta.bci	<i>Parametric bootstrap confidence interval for the parameter theta for Liouville copula</i>
-----------	--

---

## Description

The parametric bootstrap provides confidence intervals by repeatedly sampling datasets from the postulated Liouville copula model. If  $d = 2$  and the model is either `gumbel` or `clayton`, the value of Kendall's  $\tau$  is calculated from the sample, and the confidence interval or the quantiles correspond to the inverse  $\tau^{-1}(\tau(\theta))$  for the bootstrap quantile values of  $\tau$  (using monotonicity).

## Usage

```
theta.bci(
  B = 1999,
  family,
  alphavec,
  n,
  theta.hat,
  quant = c(0.025, 0.975),
  silent = FALSE
)
```

## Arguments

<code>B</code>	number of bootstrap replicates
<code>family</code>	family of the Liouville copula. Either "clayton", "gumbel", "frank", "AMH" or "joe"
<code>alphavec</code>	vector of Dirichlet allocations (must be a vector of integers)
<code>n</code>	sample size
<code>theta.hat</code>	estimate of theta
<code>quant</code>	if the vector of probability is specified, the function will return the corresponding bootstrap quantiles
<code>silent</code>	boolean for output progress. Default is FALSE, which means iterations are printed if $d > 2$ .

## Details

Install package `wdm` to speed up calculation of Kendall's tau.

Since no closed-form formulas exist for the other models or in higher dimension, the method is extremely slow since it relies on maximization of a new sample from the model and look up the corresponding parameters.

**Value**

a list with a 95 and the bootstrap values of Kendall's tau in `boot_tau` if  $d = 2$  and the model is either `gumbel` or `clayton`. Otherwise, the list contains `boot_theta`.

**Examples**

```
## Not run:  
theta.bci(B=99, family="gumbel", alphavec=c(2,3), n=100, theta.hat=2)  
theta.bci(B=19, family="AMH", alphavec=c(1,2), n=100, theta.hat=0.5, quant=c(0.05,0.95))  
theta.bci(B=19, family="frank", alphavec=c(1,2,3), n=100, theta.hat=0.5, quant=c(0.05,0.95))  
  
## End(Not run)
```

# Index

- \* **copula**
  - lcopula-package, 2
- \* **datasets**
  - airquality, 6
  - danube, 7
  - nutrient, 16
  - .ctspecdens, 3
  - .liouv.Tau\_s, 4
  - .liouv.iTau\_s, 4
  - .pickands.dir.uni, 5
  - .pickands.fun.uni, 5
  - .pliouv.opt\_old, 6
- airquality, 6
- copula, 3
- danube, 7
- dirspecdens, 8
- dliouv (Liouville), 13
- dliouvm (Liouville\_marginal), 14
- hmvevdliouv, 8
- isliouv\_m, 9
- isliouvm, 9, 14
- isliouvm (Liouville\_marginal), 14
- K.plot, 10
- lcopula (lcopula-package), 2
- lcopula-package, 2
- liouv.iTau, 11
- liouv.maxim, 11
- liouv.Tau, 12
- Liouville, 13
- Liouville\_marginal, 14, 14
- negdirspecdens, 15
- nutrient, 16
- pickands.liouv, 17
- pickands.plot, 18
- pliouv, 20
- pliouv (Liouville), 13
- pliouv.opt, 19
- pliouvm (Liouville\_marginal), 14
- rarchi, 20
- rliouv (Liouville), 13
- sliouv, 14, 20, 21
- sliouv\_m, 21
- sliouvm (Liouville\_marginal), 14
- theta.bci, 22