

# Package ‘gofCopula’

October 17, 2018

**Type** Package

**Title** Goodness-of-Fit Tests for Copulae

**Version** 0.2-4

**Date** 2018-10-17

**Author** Ostap Okhrin <ostap.okhrin@tu-dresden.de>,  
Simon Trimborn <simon.trimborn@nus.edu.sg>,  
Shulin Zhang <slzhang@swufe.edu.cn>,  
Qian M. Zhou <qmzhou@sfu.ca>

**Maintainer** Simon Trimborn <simon.trimborn@nus.edu.sg>

**LazyLoad** yes

**LazyData** true

**Depends** R (>= 1.9.0), copula (>= 0.999-15), foreach, parallel,  
doParallel, R.utils

**Imports** SparseGrid, numDeriv, VineCopula (>= 2.0.5), methods, stats,  
MASS, utils

**Description** Several GoF tests for Copulae are provided. A new hybrid test is implemented which supports all of the individual tests. Estimation methods for the margins are provided. All the tests support parameter estimation and predefined values. The parameters are estimated by pseudo maximum likelihood but if it fails the estimation switches automatically to inversion of Kendall's tau. All the tests support automatized parallelization of the bootstrapping tasks.

**License** GPL (>= 3)

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2018-10-17 16:00:03 UTC

## R topics documented:

gof . . . . .	2
gofADChisq . . . . .	4
gofADGamma . . . . .	4
gofCheckTime . . . . .	5

gofco	6
gofHybrid	8
gofKendallCvM	10
gofKendallKS	12
gofKernel	14
gofPIOSRn	17
gofPIOSTn	19
gofRn	21
gofRosenblattChisq	22
gofRosenblattGamma	24
gofRosenblattSnB	27
gofRosenblattSnC	29
gofSn	31
gofWhich	33
gofWhichCopula	34
gofWhite	35
IndexReturns	37

## Index 38

gof *Combining function for tests*

### Description

`gof` computes for a given dataset and based on the choices of the user either all tests for a given amount of copulae, performs for a given testset every test with all available copulae or computes for given copulae and tests all possible combinations.

### Usage

```
gof(x, priority = "tests", copula = NULL, tests = NULL, margins = "ranks",
    dispstr = "ex", M = 50, MJ = 50, param = 0.5, param.est = T, df = 4,
    df.est = T, m = 1, delta.J = 0.5, nodes.Integration = 12, m_b = 0.5,
    zeta.m = 0, b_Rn = 0.05, processes = 1)
```

### Arguments

<code>x</code>	A matrix containing the residuals of the data.
<code>priority</code>	A character string which is either "tests" or "copula". "tests" indicates that all implemented tests are performed for all copulae which the tests share. This are e.g. "normal" and "clayton". If "copula" is chosen, the tests which are able to test for "normal", "t", "frank", "gumbel" and "clayton" are performed. If one of the arguments tests or copula is not NULL, then priority doesn't affect the choice of the copulae and tests.
<code>copula</code>	A character vector which indicates the copula to test for.
<code>tests</code>	A character vector which indicates the tests to use.

margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
dispstr	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable and "un" for unstructured, see package copula.
M	The amount of bootstrap rounds to be performed by each test. Default is 1000.
MJ	Just for the test gofKernel. Size of bootstrapping sample.
param	The copulae parameters to use for each test, if it shall not be estimated.
param.est	Shall be either TRUE or FALSE. TRUE means that param will be estimated.
df	The degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
m	Length of blocks. Only necessary if the test gofPIOSTn is part of testset.
delta.J	Scaling parameter for the matrix of smoothing parameters. Only necessary if the test gofKernel is part of testset.
nodes.Integration	Number of knots of the bivariate Gauss-Legendre quadrature. Only necessary if the test gofKernel is part of testset.
m_b	The power of the statistic. Only necessary if the test gofRn is part of testset.
zeta.m	The adjustment parameter. Only necessary if the test gofRn is part of testset.
b_Rn	The bandwidth for the estimation of the first-order partial derivatives based on the empirical copula. Only necessary if the test gofRn is part of testset.
processes	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

## Details

If a character vector is given for the argument `copula` and nothing for `tests`, then all tests are performed for which the given copulae are implemented. If `tests` contains a character vector of tests and `copula = NULL`, then this tests will be performed for all implemented copulae. If character vectors are given for `copula` and `tests`, then the tests are performed with the given copulae. If `tests = NULL` and `copula = NULL`, then the argument `priority` catches in and defines the procedure.

The time to compute the entire procedure is always estimated in case that `M` or `MJ` are 100 or higher.

For small values of `M`, initializing the parallelization via `processes` does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelization just for high values of `M`.

**Value**

A list containing several objects of class `gofCOP` with the following components for each copulae

<code>method</code>	a character which informs about the performed analysis
<code>erg. tests</code>	a matrix with the p-values and test statistics of the individual tests including the hybrid test

**Examples**

```
data(IndexReturns)
```

```
gof(IndexReturns[c(1:100),c(1:2)], priority = "tests", copula = "normal",
tests = c("gofRosenblattSnB", "gofRosenblattSnC"), M = 10)
```

---

<code>gofADChisq</code>	<i>Renamed to <code>gofRosenblattChisq</code></i>
-------------------------	---

---

**Description**

This test was renamed to `gofRosenblattChisq`.

**Usage**

```
gofADChisq(...)
```

**Arguments**

... Parameters supposed to be in `gofRosenblattChisq`.

---

<code>gofADGamma</code>	<i>Renamed to <code>gofRosenblattGamma</code></i>
-------------------------	---

---

**Description**

This test was renamed to `gofRosenblattGamma`.

**Usage**

```
gofADGamma(...)
```

**Arguments**

... Parameters supposed to be in `gofRosenblattGamma`.

gofCheckTime

*Combining function for tests***Description**

The computation of a gof test can take very long, especially when the number of bootstrap rounds is high. The function `gofCheckTime` computes the time which the estimation most likely take.

**Usage**

```
gofCheckTime(copula, x, test, M = 1000, MJ = 10000, print.res = T, margins = "ranks",
             dispstr = "ex", param = 0.5, param.est = T, df = 4, df.est = T,
             m = 1, delta.J = 0.5, nodes.Integration = 12, m_b = 0.5,
             zeta.m = 0, b_Rn = 0.05, processes = 1)
```

**Arguments**

copula	A character vector which indicates the copula to test for.
x	A matrix containing the data.
test	A character vector which indicates the test to use.
M	The number of bootstrapping rounds which shall be later taken in the estimation.
MJ	Just for the test <code>gofKernel</code> . Size of bootstrapping sample.
print.res	Logical which defines if the resulting time shall be printed or given as value. Default is TRUE.
margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
dispstr	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable, "ar1" for AR(1), "toep" for Toeplitz, and "un" for unstructured, see package copula.
param	The copulae parameters to use for each test, if it shall not be estimated.
param.est	Shall be either TRUE or FALSE. TRUE means that param will be estimated.
df	The degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
m	Length of blocks. Only necessary if the test <code>gofPIOSTn</code> is part of testset.
delta.J	Scaling parameter for the matrix of smoothing parameters. Only necessary if the test <code>gofKernel</code> is part of testset.

nodes.Integration	Number of knots of the bivariate Gauss-Legendre quadrature. Only necessary if the test <code>gofKernel</code> is part of <code>testset</code> .
m_b	The power of the statistic. Only necessary if the test <code>gofRn</code> is part of <code>testset</code> .
zeta.m	The adjustment parameter. Only necessary if the test <code>gofRn</code> is part of <code>testset</code> .
b_Rn	The bandwidth for the estimation of the first-order partial derivatives based on the empirical copula. Only necessary if the test <code>gofRn</code> is part of <code>testset</code> .
processes	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors.

### Details

The function estimates the time which the entire `gof` test will take.

### Examples

```
data(IndexReturns)

gofCheckTime("normal", IndexReturns[c(1:50),c(1:2)], "gofRosenblattSnC", M = 10000)
```

---

gofco

*Interface with copula class*

---

### Description

`gofco` is an interface with the `copula` package. It reads out the information from a `copula` class object and hands them over to a specified `gof` test or set of tests.

### Usage

```
gofco(copulaobject, x, testset = c("gofPIOSRn", "gofKernel"), margins = "ranks",
      dispstr = "ex", M = 1000, execute.times.comp = T, m = 1, MJ = 100,
      delta.J = 0.5, nodes.Integration = 12, m_b = 0.5, zeta.m = 0, b_Rn = 0.05,
      processes = 1)
```

### Arguments

copulaobject	An object with the <code>copula</code> class from the <code>copula</code> package.
x	A matrix containing the residuals of the data. Take into account that the dimension of the matrix must fit the required dimensions from the single tests.
testset	The tests to be used. Possible are <code>"gofPIOSRn"</code> , <code>"gofPIOSTn"</code> , <code>"gofKernel"</code> , <code>"gofRosenblattSnB"</code> , <code>"gofRosenblattSnC"</code> , <code>"gofRosenblattChisq"</code> , <code>"gofRosenblattGamma"</code> , <code>"gofSn"</code> , <code>"gofKendallCvM"</code> , <code>"gofKendallKS"</code> , <code>"gofWhite"</code> , <code>"gofRn"</code> .

<code>margins</code>	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
<code>dispstr</code>	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable, "ar1" for AR(1), "toep" for Toeplitz, and "un" for unstructured, see package copula.
<code>M</code>	Number of bootstrapping samples in the single tests.
<code>execute.times.comp</code>	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.
<code>m</code>	Length of blocks. Only necessary if the test <code>gofPIOSTn</code> is part of testset.
<code>MJ</code>	Size of bootstrapping sample. Only necessary if the test <code>gofKernel</code> is part of testset.
<code>delta.J</code>	Scaling parameter for the matrix of smoothing parameters. Only necessary if the test <code>gofKernel</code> is part of testset.
<code>nodes.Integration</code>	Number of knots of the bivariate Gauss-Legendre quadrature. Only necessary if the test <code>gofKernel</code> is part of testset.
<code>m_b</code>	The power of the statistic. Only necessary if the test <code>gofRn</code> is part of testset.
<code>zeta.m</code>	The adjustment parameter. Only necessary if the test <code>gofRn</code> is part of testset.
<code>b_Rn</code>	The bandwidth for the estimation of the first-order partial derivatives based on the empirical copula. Only necessary if the test <code>gofRn</code> is part of testset.
<code>processes</code>	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors.

## Details

The function reads out the arguments in the copula class object. If the dependence parameter is not specified in the object, it is estimated. In case that the object describes a "t"-copula, then the same holds for the degrees of freedom. The dimension is not extracted from the object. It is obtained from the inserted dataset.

When more than one test shall be performed, the hybrid test is computed too.

## Value

A object of the class `gofCOP` with the components

<code>method</code>	a character which informs about the performed analysis
<code>erg.tests</code>	a matrix with the p-values and test statistics of the hybrid and the individual tests

## References

Yan, Jun. Enjoy the joy of copulas: with a package copula. Journal of Statistical Software 21.4 (2007): 1-21.

**Examples**

```
data(IndexReturns)
copObject = normalCopula()

gofco(copObject, x = IndexReturns[c(1:100),c(1:2)], testset = c("gofPIOSRn", "gofKernel"), M = 20)
```

---

gofHybrid	<i>Hybrid gof test</i>
-----------	------------------------

---

**Description**

`gofHybrid` combines all tests in this package to perform the hybrid test presented in Zhang et al. (2015). The test gives the possibility to combine several single tests which is helpful since in different test scenarios are different tests the most powerful.

**Usage**

```
gofHybrid(copula, x, testset = c("gofPIOSRn", "gofKernel"), margins = "ranks",
  dispstr = "ex", M = 1000, execute.times.comp = T, param = 0.5,
  param.est = T, df = 4, df.est = T, m = 1, MJ = 100, delta.J = 0.5,
  nodes.Integration = 12, m_b = 0.5, zeta.m = 0, b_Rn = 0.05,
  processes = 1)
```

**Arguments**

copula	The copula to test for. Possible are "normal", "t", "clayton", "gumbel" and "frank".
x	A matrix containing the residuals of the data. Take into account that the dimension of the matrix must fit the required dimensions from the single tests.
testset	The tests to be used. Possible are "gofPIOSRn", "gofPIOSTn", "gofKernel", "gofRosenblattSnB", "gofRosenblattSnC", "gofRosenblattChisq", "gofRosenblattGamma", "gofSn", "gofKendallCvM", "gofKendallKS", "gofWhite", "gofRn".
margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
dispstr	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable and "un" for unstructured, see package copula.
M	Number of bootstrapping samples in the single tests.
execute.times.comp	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.
param	The copulae parameters to use for each test, if it shall not be estimated.



param.est	Shall be either TRUE or FALSE. TRUE means that param will be estimated.
df	The degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
m	Length of blocks. Only necessary if the test gofPIOSTn is part of testset.
MJ	Size of bootstrapping sample. Only necessary if the test gofKernel is part of testset.
delta.J	Scaling parameter for the matrix of smoothing parameters. Only necessary if the test gofKernel is part of testset.
nodes.Integration	Number of knots of the bivariate Gauss-Legendre quadrature. Only necessary if the test gofKernel is part of testset.
m_b	The power of the statistic. Only necessary if the test gofRn is part of testset.
zeta.m	The adjustment parameter. Only necessary if the test gofRn is part of testset.
b_Rn	The bandwidth for the estimation of the first-order partial derivatives based on the empirical copula. Only necessary if the test gofRn is part of testset.
processes	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

## Details

In most of scenarios for goodness-of-fit tests, including the one for copula models (e.g. Genest et al. (2009)) there exists no single dominate optimal test. Zhang et al. (2015) proposed a hybrid test which performed in their simulation study more desirably compared to the applied single tests.

The p-value is a combination of the single tests in the following way:

$$p_n^{hybrid} = \min(q \cdot \min(p_n^{(1)}, \dots, p_n^{(q)}), 1)$$

where  $q$  is the number of tests and  $p_n^{(i)}$  the p-value of the test  $i$ . It is ensured that the hybrid test is consistent as long as at least one of the tests is consistent.

The computation of the individual p-values is performed as described in the details of this tests. Note that the derivation differs.

For small values of M, initializing the parallization via processes does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallization just for high values of M.

## Value

A object of the class gofCOP with the components

method	a character which informs about the performed analysis
erg.tests	a matrix with the p-values and test statistics of the hybrid and the individual tests

## References

Zhang, S., Okhrin, O., Zhou, Q., and Song, P.. Goodness-of-fit Test For Specification of Semiparametric Copula Dependence Models. *Journal of Econometrics*, 193, 2016, pp. 215-233 [dx.doi.org/10.1016/j.jeconom.2016.02.017](https://doi.org/10.1016/j.jeconom.2016.02.017)

Christian Genest, Bruno Remillard, David Beaudoin (2009). Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and Economics*, Volume 44, Issue 2, April 2009, Pages 199-213, ISSN 0167-6687. <http://dx.doi.org/10.1016/j.insmatheco.2007.10.005>

## Examples

```
data(IndexReturns)

gofHybrid("normal", IndexReturns[c(1:100),c(1:2)],
testset = c("gofRosenblattSnB", "gofRosenblattSnC"), M = 10)
```

---

gofKendallCvM	<i>gof test (Cramer-von Mises) based on Kendall's process</i>
---------------	---

---

## Description

`gofKendallCvM` tests a given dataset for a copula based on Kendall's process with the Cramer-von Mises test statistic. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. The possible copulae are "normal", "t", "gumbel", "clayton" and "frank". See for reference Genest et al. (2009). The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used. The approximate p-values are computed with a parametric bootstrap, which computation can be accelerated by enabling in-build parallel computation.

## Usage

```
gofKendallCvM(copula, x, param = 0.5, param.est = T, df = 4, df.est = T,
  margins = "ranks", dispstr = "ex", M = 100,
  execute.times.comp = T, processes = 1)
```

## Arguments

copula	The copula to test for. Possible are the copulae "normal", "t", "clayton", "gumbel" and "frank".
x	A matrix containing the residuals of the data.
param	The copula parameter to use, if it shall not be estimated.
param.est	Shall be either TRUE or FALSE. TRUE means that param will be estimated.
df	Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.

margins	Specifies which estimation method shall be used in case that the input data are not in the range $[0,1]$ . The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
dispstr	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable and "un" for unstructured, see package copula.
M	Number of bootstrap samples.
execute.times.comp	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.
processes	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

## Details

With the pseudo observations  $U_{ij}$  for  $i = 1, \dots, n$ ,  $j = 1, \dots, d$  and  $\mathbf{u} \in [0, 1]^d$  is the empirical copula given by  $C_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^n \mathbf{I}(U_{i1} \leq u_1, \dots, U_{id} \leq u_d)$ . Let the rescaled pseudo observations be  $V_1 = C_n(U_1), \dots, V_n = C_n(U_n)$  and the distribution function of  $V$  shall be  $K$ . The estimated version is given by

$$K_n(v) = \frac{1}{n} \sum_{i=1}^n \mathbf{I}(V_i \leq v)$$

with  $v \in [0, 1]^d$ . The testable  $H'_0$  hypothesis is then

$$K \in \mathcal{K}_0 = \{K_\theta : \theta \in \Theta\}$$

with  $\Theta$  being an open subset of  $R^p$  for an integer  $p \geq 1$ , see Genest et al. (2009). The resulting Cramer-von Mises test statistic is then given by

$$T = n \int_0^1 (K_n(v) - K_{\theta_n})^2 dK_{\theta_n}(v).$$

Because  $H'_0$  consists of more distributions than the  $H_0$  is the test not necessarily consistent.

The approximate p-value is computed by the formula

$$\sum_{b=1}^M \mathbf{I}(|T_b| \geq |T|) / M,$$

For small values of M, initializing the parallization via processes does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelization just for high values of M.

**Value**

A object of the class `gofCOP` with the components

`method` a character which informs about the performed analysis  
`erg.tests` a matrix with the p-value and test statistic of test

**References**

Christian Genest, Bruno Remillard, David Beaudoin (2009). Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and Economics*, Volume 44, Issue 2, April 2009, Pages 199-213, ISSN 0167-6687. <http://dx.doi.org/10.1016/j.insmatheco.2007.10.005>

Christian Genest, Jean-Francois Quessy, Bruno Remillard (2002). Tests of serial independence based on Kendall's process. *The Canadian Journal of Statistics*, Volume 30, Issue 3, Sep. 2002, Pages 441-461. <https://www.jstor.org/stable/3316147>

cp. Ulf Schepsmeier, Jakob Stoeber, Eike Christian Brechmann, Benedikt Graeler (2015). VineCopula: Statistical Inference of Vine Copulas. *R package version 1.4.* <https://cran.r-project.org/package=VineCopula>

**Examples**

```
data(IndexReturns)

gofKendallCvM("normal", IndexReturns[c(1:100),c(1:2)], M = 10)
```

---

<code>gofKendallKS</code>	<i>gof test (Kolmogorov-Smirnof) based on Kendall's process</i>
---------------------------	---

---

**Description**

`gofKendallKS` tests a given dataset for a copula based on Kendall's process with the Kolmogorov-Smirnof test statistic. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. The possible copulae are "normal", "t", "gumbel", "clayton" and "frank". See for reference Genest et al. (2009). The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used. The approximate p-values are computed with a parametric bootstrap, which computation can be accelerated by enabling in-build parallel computation.

**Usage**

```
gofKendallKS(copula, x, param = 0.5, param.est = T, df = 4, df.est = T,
             margins = "ranks", dispstr = "ex", M = 100,
             execute.times.comp = T, processes = 1)
```

**Arguments**

copula	The copula to test for. Possible are the copulae "normal", "t", "clayton", "gumbel" and "frank".
x	A matrix containing the residuals of the data.
param	The copula parameter to use, if it shall not be estimated.
param.est	Shall be either TRUE or FALSE. TRUE means that param will be estimated.
df	Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
dispstr	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable and "un" for unstructured, see package copula.
M	Number of bootstrap samples.
execute.times.comp	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.
processes	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

**Details**

With the pseudo observations  $U_{ij}$  for  $i = 1, \dots, n, j = 1, \dots, d$  and  $\mathbf{u} \in [0, 1]^d$  is the empirical copula given by  $C_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^n \mathbf{I}(U_{i1} \leq u_1, \dots, U_{id} \leq u_d)$ . Let the rescaled pseudo observations be  $V_1 = C_n(U_1), \dots, V_n = C_n(U_n)$  and the distribution function of  $V$  shall be  $K$ . The estimated version is given by

$$K_n(v) = \frac{1}{n} \sum_{i=1}^n \mathbf{I}(V_i \leq v)$$

with  $v \in [0, 1]^d$ . The testable  $H'_0$  hypothesis is then

$$K \in \mathcal{K}_0 = \{K_\theta : \theta \in \Theta\}$$

with  $\Theta$  being an open subset of  $R^p$  for an integer  $p \geq 1$ , see Genest et al. (2009). The resulting Cramer-von Mises test statistic is then given by

$$T = \sqrt{n} \sup_{v \in [0,1]} |K_n(v) - K_{\theta_n}|.$$

Because  $H'_0$  consists of more distributions than the  $H_0$  is the test not necessarily consistent.

The approximate p-value is computed by the formula

$$\sum_{b=1}^M \mathbf{I}(|T_b| \geq |T|) / M,$$

For small values of M, initializing the parallelization via processes does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelization just for high values of M.

### Value

A object of the class `gofCOP` with the components

method            a character which informs about the performed analysis  
 erg. tests        a matrix with the p-value and test statistic of test

### References

Christian Genest, Bruno Remillard, David Beaudoin (2009). Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and Economics*, Volume 44, Issue 2, April 2009, Pages 199-213, ISSN 0167-6687. <http://dx.doi.org/10.1016/j.insmatheco.2007.10.005>

Christian Genest, Jean-Francois Quessy, Bruno Remillard (2002). Tests of serial independence based on Kendall's process. *The Canadian Journal of Statistics*, Volume 30, Issue 3, Sep. 2002, Pages 441-461. <https://www.jstor.org/stable/3316147>

cp. Ulf Schepsmeier, Jakob Stoeber, Eike Christian Brechmann, Benedikt Graeler (2015). VineCopula: Statistical Inference of Vine Copulas. *R package version 1.4.*. <https://cran.r-project.org/package=VineCopula>

### Examples

```
data(IndexReturns)

gofKendallKS("normal", IndexReturns[c(1:100),c(1:2)], M = 10)
```

---

gofKernel

2 and 3 dimensional gof test of Scaillet

---

### Description

`gofKernel` tests a 2 or 3 dimensional dataset with the Scaillet test for a copula. The possible copulae are "normal", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used. The approximate p-values are computed with a parametric bootstrap, which computation can be accelerated by enabling in-build parallel computation.

**Usage**

```
gofKernel(copula, x, M = 1000, param = 0.5, param.est = T, df = 4, df.est = T,
          margins = "ranks", dispstr = "ex", MJ = 100, delta.J = 0.5,
          nodes.Integration = 12, execute.times.comp = T, processes = 1)
```

**Arguments**

copula	The copula to test for. Possible are the copulae "normal", "t", "clayton", "gumbel" and "frank".
x	A 2 or 3 dimensional matrix containing the residuals of the data.
M	Number of bootstrapping loops.
param	The parameter to be used.
param.est	Shall be either TRUE or FALSE. TRUE means that param will be estimated with a maximum likelihood estimation.
df	Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
dispstr	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable and "un" for unstructured, see package copula.
MJ	Size of bootstrapping sample.
delta.J	Scaling parameter for the matrix of smoothing parameters.
nodes.Integration	Number of knots of the bivariate Gauss-Legendre quadrature.
execute.times.comp	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.
processes	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

**Details**

The Scaillet test is a kernel-based goodness-of-fit test with a fixed smoothing parameter. For the copula density  $c(\mathbf{u}, \theta)$ , the corresponding kernel estimator is given by

$$c_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^n K_H[\mathbf{u} - (U_{i1}, \dots, U_{id})^\top],$$

where  $U_{ij}$  for  $i = 1, \dots, n; j = 1, \dots, d$  are the pseudo observations,  $\mathbf{u} \in [0, 1]^d$  and  $K_H(y) = K(H^{-1}y)/\det(H)$  for which a bivariate quadratic kernel is used, as in Scaillet (2007). The matrix of smoothing parameters is  $H = 2.6073n^{-1/6}\hat{\Sigma}^{1/2}$  with  $\hat{\Sigma}$  the sample covariance matrix. The test statistic is then given by

$$T = \int_{[0,1]^d} \{c_n(\mathbf{u}) - K_H * c(\mathbf{u}, \theta_n)\} \omega(\mathbf{u}) d\mathbf{u},$$

where  $*$  denotes the convolution operator and  $\omega$  is a weight function, see Zhang et al. (2015). The bivariate Gauss-Legendre quadrature method is used to compute the integral in the test statistic numerically, see Scaillet (2007).

The approximate p-value is computed by the formula

$$\sum_{b=1}^M \mathbf{I}(|T_b| \geq |T|) / M,$$

For small values of M, initializing the parallelization via processes does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelization just for high values of M.

## Value

A object of the class `gofCOP` with the components

<code>method</code>	a character which informs about the performed analysis
<code>statistic</code>	value of the test statistic
<code>p.value</code>	the approximate p-value

## References

Zhang, S., Okhrin, O., Zhou, Q., and Song, P. Goodness-of-fit Test For Specification of Semiparametric Copula Dependence Models. *under revision in Journal of Econometrics from 15.01.2014*  
<http://sfb649.wiwi.hu-berlin.de/papers/pdf/SFB649DP2013-041.pdf>

Scaillet, O. (2007). Kernel based goodness-of-fit tests for copulas with fixed smoothing parameters. *Journal of Multivariate Analysis*, 98:533-543

## Examples

```
data(IndexReturns)
```

```
gofKernel("normal", IndexReturns[c(1:100),c(1:2)], M = 5, MJ = 5)
```



---

gofPIOSRn	<i>2 and 3 dimensional gof test based on the in-and-out-of-sample approach</i>
-----------	--

---

### Description

gofPIOSRn tests a 2 or 3 dimensional dataset with the approximate PIOS test for a copula. The possible copulae are "normal", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used. The approximate p-values are computed with a semiparametric bootstrap, which computation can be accelerated by enabling in-build parallel computation.

### Usage

```
gofPIOSRn(copula, x, M = 1000, param = 0.5, param.est = T, df = 4, df.est = T,
  margins = "ranks", dispstr = "ex", execute.times.comp = T,
  processes = 1)
```

### Arguments

copula	The copula to test for. Possible are the copulae "normal", "t", "clayton", "gumbel" and "frank".
x	A 2 or 3 dimensional matrix containing the residuals of the data.
M	Number of bootstrapping loops.
param	The parameter to be used.
param.est	Shall be either TRUE or FALSE. TRUE means that param will be estimated with a maximum likelihood estimation.
df	Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
dispstr	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable and "un" for unstructured, see package copula.
execute.times.comp	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.
processes	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

### Details

The "Rn" test is introduced in Zhang et al. (2015). It is a information ratio statistic which is approximately equivalent to the "Tn" test, which is the PIOS test. Both test the  $H_0$  hypothesis

$$H_0 : C_0 \in \mathcal{C}.$$

"Rn" is introduced because the "Tn" test has to estimate  $n/m$  parameters which can be computationally demanding. The test statistic of the "Tn" test is defined as

$$T = \sum_{b=1}^M \sum_{k=1}^m [l\{U_k^b; \theta_n\} - l\{U_k^b; \theta_n^{-b}\}]$$

with  $l$  the log likelihood function, the pseudo observations  $U_{ij}$  for  $i = 1, \dots, n; j = 1, \dots, d$  and

$$\theta_n = \arg \min_{\theta} \sum_{i=1}^n l(U_i; \theta)$$

and

$$\theta_n^{-b} = \arg \min_{\theta} \sum_{b' \neq b}^M \sum_{k=1}^m l(U_k^{b'}; \theta), b = 1, \dots, M.$$

By defining two information matrices

$$S(\theta) = -E_0 \left[ \frac{\partial^2}{\partial \theta \partial \theta^\top} l\{U_1; \theta\} \right],$$

$$V(\theta) = -E_0 \left[ \frac{\partial}{\partial \theta} l\{U_1; \theta\} l^\top \{U_1; \theta\} \right]$$

where  $S(\cdot)$  represents the negative sensitivity matrix,  $V(\cdot)$  the variability matrix and  $E_0$  is the expectation under the true copula  $C_0$ . Under suitable regularity conditions, given in Zhang et al. (2015), holds then in probability, that

$$T = tr\{S(\theta^*)^{-1} - V(\theta^*)\}$$

as  $n \rightarrow \infty$ .

The approximate p-value is computed by the formula

$$\sum_{b=1}^M \mathbf{I}(|T_b| \geq |T|) / M,$$

For more details, see Zhang et al. (2015). The applied estimation method is the two-step pseudo maximum likelihood approach, see Genest and Rivest (1995).

For small values of M, initializing the parallization via processes does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallization just for high values of M.

**Value**

A object of the class gofCOP with the components

method	a character which informs about the performed analysis
statistic	value of the test statistic
p.value	the approximate p-value

**References**

Zhang, S., Okhrin, O., Zhou, Q., and Song, P.. Goodness-of-fit Test For Specification of Semiparametric Copula Dependence Models. *under revision in Journal of Econometrics from 15.01.2014*  
<http://sfb649.wiwi.hu-berlin.de/papers/pdf/SFB649DP2013-041.pdf>

Genest, C., K. G. and Rivest, L.-P. (1995). A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82:534-552

**Examples**

```
data(IndexReturns)

gofPIOSRn("normal", IndexReturns[c(1:100),c(1:2)], M = 10)
```

---

gofPIOSTn	<i>2 and 3 dimensional gof test based on the in-and-out-of-sample approach</i>
-----------	--

---

**Description**

gofPIOSTn tests a 2 or 3 dimensional dataset with the PIOS test for a copula. The possible copulae are "normal", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used. The approximate p-values are computed with a semiparametric bootstrap, which computation can be accelerated by enabling in-build parallel computation.

**Usage**

```
gofPIOSTn(copula, x, M = 1000, param = 0.5, param.est = T, df = 4, df.est = T,
  margins = "ranks", dispstr = "ex", m = 1,
  execute.times.comp = T, processes = 1)
```

**Arguments**

copula	The copula to test for. Possible are the copulae "normal", "t", "clayton", "gumbel" and "frank".
x	A 2 or 3 dimensional matrix containing the residuals of the data.
M	Number of bootstrapping loops.

param	The parameter to be used.
param.est	Shall be either TRUE or FALSE. TRUE means that param will be estimated with a maximum likelihood estimation.
df	Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
dispstr	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable and "un" for unstructured, see package copula.
m	Length of blocks.
execute.times.comp	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.
processes	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

## Details

The "Tn" test is introduced in Zhang et al. (2015). It tests the  $H_0$  hypothesis

$$H_0 : C_0 \in \mathcal{C}.$$

For the test are constructed blocks of length  $m$  out of the data. The test compares then the pseudo likelihood of the data in each block with the overall parameter and with the parameter by leaving out the data in the block. By this procedure can be determined if the data in the block influence the parameter estimation significantly. The test statistic is defined as

$$T = \sum_{b=1}^M \sum_{k=1}^m [l\{U_k^b; \theta_n\} - l\{U_k^b; \theta_n^{-b}\}]$$

with the pseudo observations  $U_{ij}$  for  $i = 1, \dots, n; j = 1, \dots, d$  and

$$\theta_n = \arg \min_{\theta} \sum_{i=1}^n l(U_i; \theta)$$

and

$$\theta_n^{-b} = \arg \min_{\theta} \sum_{b' \neq b}^M \sum_{i=1}^m l(U_i^{b'}; \theta), b = 1, \dots, M.$$

The approximate p-value is computed by the formula

$$\sum_{b=1}^M \mathbf{I}(|T_b| \geq |T|) / M,$$

The applied estimation method is the two-step pseudo maximum likelihood approach, see Genest and Rivest (1995).

For small values of M, initializing the parallization via processes does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelization just for high values of M.

### Value

A object of the class `gofCOP` with the components

<code>method</code>	a character which informs about the performed analysis
<code>statistic</code>	value of the test statistic
<code>p.value</code>	the approximate p-value

### References

Zhang, S., Okhrin, O., Zhou, Q., and Song, P. Goodness-of-fit Test For Specification of Semiparametric Copula Dependence Models. *under revision in Journal of Econometrics from 15.01.2014*  
<http://sfb649.wiwi.hu-berlin.de/papers/pdf/SFB649DP2013-041.pdf>

Genest, C., K. G. and Rivest, L.-P. (1995). A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82:534-552

### Examples

```
data(IndexReturns)

gofPIOSTn("normal", IndexReturns[c(1:100),c(1:2)], M = 10)
```

---

<code>gofRn</code>	<i>Removed due to inconsistencies with the remaining tests</i>
--------------------	--

---

### Description

This test was removed due to inconsistencies with the remaining tests. If you still like to run it, please see the package `copula`.

### Usage

```
gofRn(...)
```

### Arguments

... Parameters previously used in `gofRn`.

---

gofRosenblattChisq      *Gof test using the Anderson-Darling test statistic and the chi-square distribution*

---

## Description

`gofRosenblattChisq` contains the RosenblattChisq gof test for copulae, described in Genest (2009) and Hofert (2014), and compares the empirical copula against a parametric estimate of the copula derived under the null hypothesis. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. The approximate p-values are computed with a parametric bootstrap, which computation can be accelerated by enabling in-build parallel computation. The gof statistics are computed with the function `gofTstat` from the package `copula`. It is possible to insert datasets of all dimensions above 1 and the possible copulae are "normal", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

## Usage

```
gofRosenblattChisq(copula, x, M = 1000, param = 0.5, param.est = T, df = 4,
  df.est = T, margins = "ranks", dispstr = "ex",
  execute.times.comp = T, processes = 1)
```

## Arguments

<code>copula</code>	The copula to test for. Possible are "normal", "t", "clayton", "gumbel" and "frank".
<code>x</code>	A matrix containing the residuals of the data.
<code>M</code>	Number of bootstrapping loops.
<code>param</code>	The copula parameter to use, if it shall not be estimated.
<code>param.est</code>	Shall be either TRUE or FALSE. TRUE means that param will be estimated.
<code>df</code>	Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
<code>df.est</code>	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
<code>margins</code>	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
<code>dispstr</code>	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable and "un" for unstructured, see package <code>copula</code> .
<code>execute.times.comp</code>	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.

processes      The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

### Details

As written in Hofert et al. (2014) computes this Anderson-Darling test statistic (supposedly)  $U[0,1]$ -distributed (under  $H_0$ ) random variates via the distribution function of chi-square distribution with  $d$  degrees of freedom. The  $H_0$  hypothesis is

$$C \in \mathcal{C}_0$$

with  $\mathcal{C}_0$  as the true class of copulae under  $H_0$ .

This test is based on the Rosenblatt probability integral transform which uses the mapping  $\mathcal{R} : (0, 1)^d \rightarrow (0, 1)^d$ . Following Genest et al. (2009) ensures this transformation the decomposition of a random vector  $\mathbf{u} \in [0, 1]^d$  with a distribution into mutually independent elements with a uniform distribution on the unit interval. The mapping provides pseudo observations  $E_i$ , given by

$$E_1 = \mathcal{R}(U_1), \dots, E_n = \mathcal{R}(U_n).$$

The mapping is performed by assigning to every vector  $\mathbf{u}$  for  $e_1 = u_1$  and for  $i \in \{2, \dots, d\}$ ,

$$e_i = \frac{\partial^{i-1} C(u_1, \dots, u_i, 1, \dots, 1)}{\partial u_1 \cdots \partial u_{i-1}} / \frac{\partial^{i-1} C(u_1, \dots, u_{i-1}, 1, \dots, 1)}{\partial u_1 \cdots \partial u_{i-1}}.$$

The Anderson-Darling test statistic of the variates

$$G(x_j) = \chi_d^2(x_j)$$

is computed (via `ADGofTest::ad.test`), where  $x_j = \sum_{i=1}^d (\Phi^{-1}(e_{ij}))^2$ ,  $\Phi^{-1}$  denotes the quantile function of the standard normal distribution function,  $\chi_d^2$  denotes the distribution function of the chi-square distribution with  $d$  degrees of freedom, and  $u_{ij}$  is the  $j$ th component in the  $i$ th row of  $\mathbf{u}$ .

The test statistic is then given by

$$T = -n - \sum_{j=1}^n \frac{2j-1}{n} [\ln(G(x_j)) + \ln(1 - G(x_{n+1-j}))].$$

The approximate p-value is computed by the formula,

$$\sum_{b=1}^M \mathbf{I}(|T_b| \geq |T|) / M,$$

where  $T$  and  $T_b$  denote the test statistic and the bootstrapped test statistic, respectively.

For small values of  $M$ , initializing the parallelization via `processes` does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelization just for high values of  $M$ .

**Value**

A object of the class `gofCOP` with the components

`method` a character which informs about the performed analysis  
`erg.tests` a matrix with the p-value and test statistic of test

**References**

Christian Genest, Bruno Remillard, David Beaudoin (2009). Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and Economics*, Volume 44, Issue 2, April 2009, Pages 199-213, ISSN 0167-6687. <http://dx.doi.org/10.1016/j.insmatheco.2007.10.005>

Marius Hofert, Ivan Kojadinovic, Martin Maechler, Jun Yan (2014). copula: Multivariate Dependence with Copulas. *R package version 0.999-15.*. <https://cran.r-project.org/package=copula>

**Examples**

```
data(IndexReturns)

gofRosenblattChisq("normal", IndexReturns[c(1:100),c(1:2)], M = 10)
```

---

<code>gofRosenblattGamma</code>	<i>Gof test using the Anderson-Darling test statistic and the gamma distribution</i>
---------------------------------	--

---

**Description**

`gofRosenblattGamma` contains the RosenblattGamma gof tests for copulae, described in Genest (2009) and Hofert (2014), and compares the empirical copula against a parametric estimate of the copula derived under the null hypothesis. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. The approximate p-values are computed with a parametric bootstrap, which computation can be accelerated by enabling in-built parallel computation. The gof statistics are computed with the function `gofTstat` from the package `copula`. It is possible to insert datasets of all dimensions above 1 and the possible copulae are "normal", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

**Usage**

```
gofRosenblattGamma(copula, x, M = 1000, param = 0.5, param.est = T, df = 4,
  df.est = T, margins = "ranks", dispstr = "ex",
  execute.times.comp = T, processes = 1)
```



**Arguments**

copula	The copula to test for. Possible are "normal", "t", "clayton" and "gumbel".
x	A matrix containing the residuals of the data.
M	Number of bootstrapping loops.
param	The copula parameter to use, if it shall not be estimated.
param.est	Shall be either TRUE or FALSE. TRUE means that param will be estimated.
df	Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
dispstr	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable and "un" for unstructured, see package copula.
execute.times.comp	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.
processes	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

**Details**

As written in Hofert et al. (2014) computes this Anderson-Darling test statistic for (supposedly)  $U[0,1]$ -distributed (under  $H_0$ ) random variates via the distribution function of the gamma distribution. The  $H_0$  hypothesis is

$$C \in \mathcal{C}_0$$

with  $\mathcal{C}_0$  as the true class of copulae under  $H_0$ .

This test is based on the Rosenblatt probability integral transform which uses the mapping  $\mathcal{R} : (0, 1)^d \rightarrow (0, 1)^d$ . Following Genest et al. (2009) ensures this transformation the decomposition of a random vector  $\mathbf{u} \in [0, 1]^d$  with a distribution into mutually independent elements with a uniform distribution on the unit interval. The mapping provides pseudo observations  $E_i$ , given by

$$E_1 = \mathcal{R}(U_1), \dots, E_n = \mathcal{R}(U_n).$$

The mapping is performed by assigning to every vector  $\mathbf{u}$  for  $e_1 = u_1$  and for  $i \in \{2, \dots, d\}$ ,

$$e_i = \frac{\partial^{i-1} C(u_1, \dots, u_i, 1, \dots, 1)}{\partial u_1 \cdots \partial u_{i-1}} / \frac{\partial^{i-1} C(u_1, \dots, u_{i-1}, 1, \dots, 1)}{\partial u_1 \cdots \partial u_{i-1}}.$$

The Anderson-Darling test statistic of the variates

$$G(x_j) = \Gamma_d(x_j)$$

is computed (via `ADGofTest::ad.test`), where  $x_j = \sum_{i=1}^d (-\ln e_{ij})$ ,  $\Gamma_d()$  denotes the distribution function of the gamma distribution with shape parameter  $d$  and shape parameter one (being equal to an Erlang( $d$ ) distribution function).

The test statistic is then given by

$$T = -n - \sum_{j=1}^n \frac{2j-1}{n} [\ln(G(x_j)) + \ln(1 - G(x_{n+1-j}))].$$

The approximate p-value is computed by the formula,

$$\sum_{b=1}^M \mathbf{I}(|T_b| \geq |T|) / M,$$

where  $T$  and  $T_b$  denote the test statistic and the bootstrapped test statistic, respectively.

For small values of  $M$ , initializing the parallelization via processes does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelization just for high values of  $M$ .

## Value

A object of the class `gofCOP` with the components

<code>method</code>	a character which informs about the performed analysis
<code>erg.tests</code>	a matrix with the p-value and test statistic of test

## References

Christian Genest, Bruno Remillard, David Beaudoin (2009). Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and Economics*, Volume 44, Issue 2, April 2009, Pages 199-213, ISSN 0167-6687. <http://dx.doi.org/10.1016/j.insmatheco.2007.10.005>

Marius Hofert, Ivan Kojadinovic, Martin Maechler, Jun Yan (2014). `copula`: Multivariate Dependence with Copulas. *R package version 0.999-15..* <https://cran.r-project.org/package=copula>

## Examples

```
data(IndexReturns)
```

```
gofRosenblattGamma("normal", IndexReturns[c(1:100),c(1:2)], M = 10)
```

**Description**

`gofRosenblattSnB` contains the SnB gof test for copulae from Genest (2009) and compares the empirical copula against a parametric estimate of the copula derived under the null hypothesis. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. The approximate p-values are computed with a parametric bootstrap, which computation can be accelerated by enabling in-build parallel computation. The gof statistics are computed with the function `gofTstat` from the package `copula`. It is possible to insert datasets of all dimensions above 1 and the possible copulae are "normal", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

**Usage**

```
gofRosenblattSnB(copula, x, M = 1000, param = 0.5, param.est = T, df = 4,
                 df.est = T, margins = "ranks", dispstr = "ex",
                 execute.times.comp = T, processes = 1)
```

**Arguments**

<code>copula</code>	The copula to test for. Possible are "normal", "t", "clayton", "gumbel" and "frank".
<code>x</code>	A matrix containing the residuals of the data.
<code>M</code>	Number of bootstrapping loops.
<code>param</code>	The copula parameter to use, if it shall not be estimated.
<code>param.est</code>	Shall be either TRUE or FALSE. TRUE means that <code>param</code> will be estimated.
<code>df</code>	Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
<code>df.est</code>	Indicates if <code>df</code> shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
<code>margins</code>	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
<code>dispstr</code>	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable and "un" for unstructured, see package <code>copula</code> .
<code>execute.times.comp</code>	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if <code>M</code> is at least 100.

processes      The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

### Details

This test is based on the Rosenblatt probability integral transform which uses the mapping  $\mathcal{R} : (0, 1)^d \rightarrow (0, 1)^d$  to test the  $H_0$  hypothesis

$$C \in \mathcal{C}_0$$

with  $\mathcal{C}_0$  as the true class of copulae under  $H_0$ . Following Genest et al. (2009) ensures this transformation the decomposition of a random vector  $\mathbf{u} \in [0, 1]^d$  with a distribution into mutually independent elements with a uniform distribution on the unit interval. The mapping provides pseudo observations  $E_i$ , given by

$$E_1 = \mathcal{R}(U_1), \dots, E_n = \mathcal{R}(U_n).$$

The mapping is performed by assigning to every vector  $\mathbf{u}$  for  $e_1 = u_1$  and for  $i \in \{2, \dots, d\}$ ,

$$e_i = \frac{\partial^{i-1} C(u_1, \dots, u_i, 1, \dots, 1)}{\partial u_1 \dots \partial u_{i-1}} / \frac{\partial^{i-1} C(u_1, \dots, u_{i-1}, 1, \dots, 1)}{\partial u_1 \dots \partial u_{i-1}}.$$

The resulting independence copula is given by  $C_{\perp}(\mathbf{u}) = u_1 \cdot \dots \cdot u_d$ .

The test statistic  $T$  is then defined as

$$T = n \int_{[0,1]^d} \{D_n(\mathbf{u}) - C_{\perp}(\mathbf{u})\}^2 d(\mathbf{u})$$

with  $D_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^n \mathbf{I}(E_i \leq \mathbf{u})$ .

The approximate p-value is computed by the formula, see **copula**,

$$\sum_{b=1}^M \mathbf{I}(|T_b| \geq |T|) / M,$$

where  $T$  and  $T_b$  denote the test statistic and the bootstrapped test statistic, respectively.

For small values of  $M$ , initializing the parallization via processes does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelization just for high values of  $M$ .

### Value

A object of the class `gofCOP` with the components

method      a character which informs about the performed analysis  
 erg. tests    a matrix with the p-value and test statistic of test

## References

Christian Genest, Bruno Remillard, David Beaudoin (2009). Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and Economics, Volume 44, Issue 2, April 2009, Pages 199-213, ISSN 0167-6687*. <http://dx.doi.org/10.1016/j.insmatheco.2007.10.005>

Marius Hofert, Ivan Kojadinovic, Martin Maechler, Jun Yan (2014). copula: Multivariate Dependence with Copulas. *R package version 0.999-15..* <https://cran.r-project.org/package=copula>

## Examples

```
data(IndexReturns)

gofRosenblattSnB("normal", IndexReturns[c(1:100),c(1:2)], M = 10)
```

---

gofRosenblattSnC      *The SnC test based on the Rosenblatt transformation*

---

## Description

`gofRosenblattSnC` contains the SnC gof test from Genest (2009) for copulae and compares the empirical copula against a parametric estimate of the copula derived under the null hypothesis. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. The approximate p-values are computed with a parametric bootstrap, which computation can be accelerated by enabling in-build parallel computation. The gof statistics are computed with the function `gofTstat` from the package `copula`. It is possible to insert datasets of all dimensions above 1 and the possible copulae are "normal", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

## Usage

```
gofRosenblattSnC(copula, x, M = 1000, param = 0.5, param.est = T, df = 4,
  df.est = T, margins = "ranks", dispstr = "ex",
  execute.times.comp = T, processes = 1)
```

## Arguments

<code>copula</code>	The copula to test for. Possible are "normal", "t", "clayton", "gumbel" and "frank".
<code>x</code>	A matrix containing the residuals of the data.
<code>M</code>	Number of bootstrapping loops.
<code>param</code>	The copula parameter to use, if it shall not be estimated.
<code>param.est</code>	Shall be either TRUE or FALSE. TRUE means that param will be estimated.

df	Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
dispstr	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable and "un" for unstructured, see package copula.
execute.times.comp	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.
processes	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

## Details

This test is based on the Rosenblatt probability integral transform which uses the mapping  $\mathcal{R} : (0, 1)^d \rightarrow (0, 1)^d$  to test the  $H_0$  hypothesis

$$C \in \mathcal{C}_0$$

with  $\mathcal{C}_0$  as the true class of copulae under  $H_0$ . Following Genest et al. (2009) ensures this transformation the decomposition of a random vector  $\mathbf{u} \in [0, 1]^d$  with a distribution into mutually independent elements with a uniform distribution on the unit interval. The mapping provides pseudo observations  $E_i$ , given by

$$E_1 = \mathcal{R}(U_1), \dots, E_n = \mathcal{R}(U_n).$$

The mapping is performed by assigning to every vector  $\mathbf{u}$  for  $e_1 = u_1$  and for  $i \in \{2, \dots, d\}$ ,

$$e_i = \frac{\partial^{i-1} C(u_1, \dots, u_i, 1, \dots, 1)}{\partial u_1 \dots \partial u_{i-1}} / \frac{\partial^{i-1} C(u_1, \dots, u_{i-1}, 1, \dots, 1)}{\partial u_1 \dots \partial u_{i-1}}.$$

The resulting independence copula is given by  $C_{\perp}(\mathbf{u}) = u_1 \cdot \dots \cdot u_d$ .

The test statistic  $T$  is then defined as

$$T = n \int_{[0,1]^d} \{D_n(\mathbf{u}) - C_{\perp}(\mathbf{u})\}^2 dD_n(\mathbf{u})$$

with  $D_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^n \mathbf{I}(E_i \leq \mathbf{u})$ .

The approximate p-value is computed by the formula, see **copula**,

$$\sum_{b=1}^M \mathbf{I}(|T_b| \geq |T|) / M,$$

where  $T$  and  $T_b$  denote the test statistic and the bootstrapped test statistic, respectively.

For small values of  $M$ , initializing the parallelization via processes does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelization just for high values of  $M$ .

### Value

A object of the class `gofCOP` with the components

`method`            a character which informs about the performed analysis  
`erg.test`            a matrix with the p-value and test statistic of test

### References

Christian Genest, Bruno Remillard, David Beaudoin (2009). Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and Economics, Volume 44, Issue 2, April 2009, Pages 199-213, ISSN 0167-6687*. <http://dx.doi.org/10.1016/j.insmatheco.2007.10.005>

Marius Hofert, Ivan Kojadinovic, Martin Maechler, Jun Yan (2014). `copula`: Multivariate Dependence with Copulas. *R package version 0.999-15..* <https://cran.r-project.org/package=copula>

### Examples

```
data(IndexReturns)

gofRosenblattSnC("normal", IndexReturns[c(1:100),c(1:2)], M = 10)
```

---

<code>gofSn</code>	<i>The Sn gof test using the empirical copula</i>
--------------------	---

---

### Description

`gofSn` performs the "Sn" gof test, described in Genest et al. (2009), for copulae and compares the empirical copula against a parametric estimate of the copula derived under the null hypothesis. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. The approximate p-values are computed with a parametric bootstrap, which computation can be accelerated by enabling in-build parallel computation. The gof statistics are computed with the function `gofTstat` from the package `copula`. It is possible to insert datasets of all dimensions above 1 and the possible copulae are "normal", "t", "gumbel", "clayton" and "frank". The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used.

### Usage

```
gofSn(copula, x, M = 1000, param = 0.5, param.est = T, df = 4, df.est = T,
      margins = "ranks", dispstr = "ex", execute.times.comp = T, processes = 1)
```

**Arguments**

copula	The copula to test for. Possible are "normal", "t", "clayton", "gumbel" and "frank".
x	A matrix containing the residuals of the data.
M	Number of bootstrapping loops.
param	The copula parameter to use, if it shall not be estimated.
param.est	Shall be either TRUE or FALSE. TRUE means that param will be estimated.
df	Degrees of freedom, if not meant to be estimated. Only necessary if tested for "t"-copula.
df.est	Indicates if df shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
margins	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
dispstr	A character string specifying the type of the symmetric positive definite matrix characterizing the elliptical copula. Implemented structures are "ex" for exchangeable and "un" for unstructured, see package copula.
execute.times.comp	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if M is at least 100.
processes	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

**Details**

With the pseudo observations  $U_{ij}$  for  $i = 1, \dots, n, j = 1, \dots, d$  and  $\mathbf{u} \in [0, 1]^d$  is the empirical copula given by  $C_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^n \mathbf{I}(U_{i1} \leq u_1, \dots, U_{id} \leq u_d)$ . It shall be tested the  $H_0$  hypothesis:

$$C \in \mathcal{C}_0$$

with  $\mathcal{C}_0$  as the true class of copulae under  $H_0$ . The test statistic  $T$  is then defined as

$$T = n \int_{[0,1]^d} \{C_n(\mathbf{u}) - C_{\theta_n}(\mathbf{u})\}^2 dC_n(\mathbf{u})$$

with  $C_{\theta_n}(\mathbf{u})$  the estimation of  $C$  under the  $H_0$ .

The approximate p-value is computed by the formula,

$$\sum_{b=1}^M \mathbf{I}(|T_b| \geq |T|) / M,$$

where  $T$  and  $T_b$  denote the test statistic and the bootstrapped test statistic, respectively.

For small values of M, initializing the parallelization via processes does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelization just for high values of M.



**Value**

A object of the class gofCOP with the components

method            a character which informs about the performed analysis  
 erg. tests        a matrix with the p-value and test statistic of test

**References**

Rosenblatt, M. (1952). Remarks on a Multivariate Transformation. *The Annals of Mathematical Statistics* 23, 3, 470-472.

Hering, C. and Hofert, M. (2014). Goodness-of-fit tests for Archimedean copulas in high dimensions. *Innovations in Quantitative Risk Management*.

Marius Hofert, Ivan Kojadinovic, Martin Maechler, Jun Yan (2014). copula: Multivariate Dependence with Copulas. *R package version 0.999-15.* <https://cran.r-project.org/package=copula>

**Examples**

```
data(IndexReturns)

gofSn("normal", IndexReturns[c(1:100),c(1:2)], M = 10)
```

---

gofWhich	<i>Applicable gof tests for testing problem</i>
----------	---

---

**Description**

`gofWhich` returns for a given copula and a dimension the applicable implemented tests.

**Usage**

```
gofWhich(copula, d)
```

**Arguments**

copula            The copula to test for. Possible are the copulae "normal", "clayton", "gumbel" and "frank".  
 d                 The dimension to search for.

**Details**

Before performing a gof test on a dataset, it pays out to have a close look at the Scatterplot to receive an idea about the possible type of copula. Afterwards follows the decision about the test. The tests in this package can be used for different types of copulae functions and dimensions. This function is dedicated to help finding the applicable gof tests for the dataset.

**Value**

A character vector which consists of the names of the tests.

**Examples**

```
gofWhich("clayton", d = 2)
```

```
gofWhich("gumbel", d = 5)
```

---

gofWhichCopula	<i>Implemented copula for a certain test</i>
----------------	--

---

**Description**

`gofWhichCopula` returns for a given test the applicable implemented copula.

**Usage**

```
gofWhichCopula(test)
```

**Arguments**

`test`            The test to search for copula.

**Details**

In case that the decision for a certain gof test was already done, it is interesting to know which copula can be used with this test.

**Value**

A character vector which consists of the names of the copula.

**Examples**

```
gofWhichCopula("gofRosenblattSnB")
```

```
gofWhichCopula("gofPIOSTn")
```

gofWhite

*2 dimensional gof tests based on White's information matrix equality.***Description**

`gofWhite` tests a given 2 dimensional dataset for a copula with the gof test based on White's information matrix equality. The possible copulae are "normal", "t", "gumbel", "clayton" and "frank". See for reference Schepsmeier et al. (2015). The parameter estimation is performed with pseudo maximum likelihood method. In case the estimation fails, inversion of Kendall's tau is used. The margins can be estimated by a bunch of distributions and the time which is necessary for the estimation can be given. The approximate p-values are computed with a parametric bootstrap, which computation can be accelerated by enabling in-build parallel computation. The computation of the test statistic and p-values is performed by corresponding functions from the [VineCopula](#) package.

**Usage**

```
gofWhite(copula, x, M = 1000, param = 0.5, param.est = T, df = 4, df.est = T,
         margins = "ranks", execute.times.comp = T, processes = 1)
```

**Arguments**

<code>copula</code>	The copula to test for. Possible are the copulae "normal", "clayton", "gumbel" and "frank".
<code>x</code>	A 2 dimensional matrix containing the residuals of the data.
<code>M</code>	Number of bootstrap samples.
<code>param</code>	The parameter to be used.
<code>param.est</code>	Shall be either TRUE or FALSE. TRUE means that <code>param</code> will be estimated with a maximum likelihood estimation.
<code>df</code>	The degrees of freedom for the "t"-copula.
<code>df.est</code>	Indicates if <code>df</code> shall be estimated. Has to be either FALSE or TRUE, where TRUE means that it will be estimated.
<code>margins</code>	Specifies which estimation method shall be used in case that the input data are not in the range [0,1]. The default is "ranks", which is the standard approach to convert data in such a case. Alternatively can the following distributions be specified: "beta", "cauchy", Chi-squared ("chisq"), "f", "gamma", Log normal ("lnorm"), Normal ("norm"), "t", "weibull", Exponential ("exp").
<code>execute.times.comp</code>	Logical. Defines if the time which the estimation most likely takes shall be computed. It'll be just given if <code>M</code> is at least 100.
<code>processes</code>	The number of parallel processes which are performed to speed up the bootstrapping. Shouldn't be higher than the number of logical processors. Please see the details.

### Details

The details are obtained from Schepsmeier et al. (2015) who states that this test uses the information matrix equality of White (1982). Under correct model specification is the Fisher Information equivalently calculated as minus the expected Hessian matrix or as the expected outer product of the score function. The null hypothesis is

$$H_0 : \mathbf{H}(\theta) + \mathbf{S}(\theta) = 0$$

where  $\mathbf{H}(\theta)$  is the expected Hessian matrix and  $\mathbf{S}(\theta)$  is the expected outer product of the score function.

The test statistic is derived by

$$T_n = n(\bar{d}(\theta_n))^\top V_{\theta_n}^{-1} \bar{d}(\theta_n)$$

with

$$\bar{d}(\theta_n) = \frac{1}{n} \sum_{i=1}^n \text{vech}(\mathbf{H}_n(\theta_n|\mathbf{u}) + \mathbf{S}_n(\theta_n|\mathbf{u})),$$

$$d(\theta_n) = \text{vech}(\mathbf{H}_n(\theta_n|\mathbf{u}) + \mathbf{S}_n(\theta_n|\mathbf{u})),$$

$$V_{\theta_n} = \frac{1}{n} \sum_{i=1}^n (d(\theta_n) - D_{\theta_n} \mathbf{H}_n(\theta_n)^{-1} \delta l(\theta_n))(d(\theta_n) - D_{\theta_n} \mathbf{H}_n(\theta_n)^{-1} \delta l(\theta_n))^\top$$

and

$$D_{\theta_n} = \frac{1}{n} \sum_{i=1}^n [\delta_{\theta_k} d_l(\theta_n)]_{l=1, \dots, \frac{p(p+1)}{2}, k=1, \dots, p}$$

where  $l(\theta_n)$  represents the log likelihood function and  $p$  is the length of the parameter vector  $\theta$ .

The test statistic will be rejected if

$$T > (1 - \alpha)(\chi_{p(p+1)/2}^2)^{-1}.$$

For small values of  $M$ , initializing the parallization via processes does not make sense. The registration of the parallel processes increases the computation time. Please consider to enable parallelization just for high values of  $M$ .

### Value

A object of the class `gofCOP` with the components

<code>method</code>	a character which informs about the performed analysis
<code>erg. tests</code>	a matrix with the p-value and test statistic of test

## References

Ulf Schepsmeier, Jakob Stoeber, Eike Christian Brechmann, Benedikt Graeler (2015). VineCopula: Statistical Inference of Vine Copulas. *R package version 1.4.* <https://cran.r-project.org/package=VineCopula>

Schepsmeier, U. and J. Stoeber (2014). Derivatives and Fisher information of bivariate copulas. *Statistical Papers*, 55(2), 525-542. <http://link.springer.com/article/10.1007/s00362-013-0498-x>

Stoeber, J. and U. Schepsmeier (2013). Estimating standard errors in regular vine copula models *Computational Statistics*, 28 (6), 2679-2707

Schepsmeier, U. (2015). Efficient information based goodness-of-fit tests for vine copula models with fixed margins. *Journal of Multivariate Analysis* 138, 34-52. Schepsmeier, U. (2014). A goodness-of-fit test for regular vine copula models.

## Examples

```
data(IndexReturns)

gofWhite("normal", IndexReturns[c(1:100),c(1:2)], M = 10)
```

---

IndexReturns

*Log returns of european stock indices.*

---

## Description

A dataset containing the log returns of 4 european stock indices in the time period 1991 - 1998.

## Usage

```
IndexReturns
```

## Format

A matrix with 1859 rows and 4 columns

## Source

The raw data are part of the R datasets and were kindly provided by Erste Bank AG, Vienna, Austria.

# Index

[gof](#), [2](#), [2](#)  
[gofADChisq](#), [4](#)  
[gofADGamma](#), [4](#)  
[gofCheckTime](#), [5](#), [5](#)  
[gofco](#), [6](#), [6](#)  
[gofHybrid](#), [8](#), [8](#)  
[gofKendallCVM](#), [10](#), [10](#)  
[gofKendallKS](#), [12](#), [12](#)  
[gofKernel](#), [14](#)  
[gofPIOSRn](#), [17](#)  
[gofPIOSTn](#), [19](#)  
[gofRn](#), [21](#)  
[gofRosenblattChisq](#), [22](#), [22](#)  
[gofRosenblattGamma](#), [24](#), [24](#)  
[gofRosenblattSnB](#), [27](#), [27](#)  
[gofRosenblattSnC](#), [29](#), [29](#)  
[gofSn](#), [31](#), [31](#)  
[gofTstat](#), [22](#), [24](#), [27](#), [29](#), [31](#)  
[gofWhich](#), [33](#), [33](#)  
[gofWhichCopula](#), [34](#), [34](#)  
[gofWhite](#), [35](#), [35](#)  
  
[IndexReturns](#), [37](#)  
  
[VineCopula](#), [35](#)